# Méthodes formelles et techniques de validation pour garantir la sécurité des systèmes automobiles

Moez Krichen

# Méthodes formelles et techniques de validation pour garantir la sécurité des systèmes automobiles

Moez Krichen

Laboratoire ReDCAD, Université de Sfax, Tunisie
moez.krichen@redcad.org

**Résumé.** La complexité croissante et la connectivité des systèmes automobiles suscitent des inquiétudes quant à leur vulnérabilité aux atteintes à la sécurité. Par conséquent, l'intégration de méthodes formelles et de techniques de validation est devenue cruciale pour garantir la sécurité des systèmes automobiles. Cet article de recherche vise à fournir un aperçu complet des méthodes formelles et des techniques de validation de pointe actuellement utilisées dans l'industrie automobile pour la sécurité des systèmes. L'article commence par discuter des défis associés à la sécurité des systèmes automobiles et des conséquences potentielles des atteintes à la sécurité. Ensuite, il explore différentes méthodes formelles, telles que la vérification de modèles, la démonstration de théorèmes et l'interprétation abstraite, qui ont été largement utilisées pour analyser et vérifier les propriétés de sécurité des systèmes automobiles. De plus, l'enquête met en évidence les techniques de validation utilisées pour garantir l'efficacité des mesures de sécurité, notamment les tests de pénétration, l'injection de fautes et les tests de fuzz. En outre, l'article examine l'intégration de méthodes formelles et de techniques de validation dans le cycle de développement automobile, y compris les phases d'ingénierie des exigences, de conception, de mise en œuvre et de test. Il aborde les avantages et les limites de ces approches, en tenant compte de facteurs tels que l'évolutivité, l'efficacité et l'applicabilité aux systèmes automobiles réels. Grâce à une revue approfondie de la littérature pertinente et d'études de cas, cette enquête offre un aperçu des tendances actuelles de la recherche, des défis et des questions de recherche ouvertes dans le domaine des méthodes formelles et des techniques de validation pour la sécurité des systèmes automobiles. Les résultats de cette enquête peuvent constituer une ressource précieuse pour les chercheurs, les praticiens et les décideurs impliqués dans la conception, le développement et l'évaluation de systèmes automobiles sécurisés.

# 1 Introduction

L'avancée rapide de la technologie des véhicules a annoncé une nouvelle ère de connectivité et de sophistication extraordinaires. Les véhicules modernes sont équipés d'unités de contrôle électronique élaborées, de réseaux intégrés et d'une pléthore de capteurs, permettant des fonctions avancées allant de la conduite autonome à des systèmes d'infodivertissement sophistiqués (211; 223; 79; 227). Bien que cet essor de compétences technologiques présente de nombreux avantages, il soulève également une préoccupation importante : la vulnérabilité des systèmes automobiles aux atteintes à la sécurité. Les véhicules sont vulnérables à divers risques cybernétiques en raison de l'intégration de logiciels complexes et d'éléments de connexion, allant du piratage à distance à l'accès illégal (10; 230). De telles atteintes à la sécurité peuvent avoir des conséquences variées, allant de la compromission des données personnelles à la mise en danger de la sécurité des passagers (242; 35). Alors que l'industrie automobile se dirige vers un avenir de véhicules autonomes et connectés, protéger ces systèmes contre les attaques malveillantes devient essentiel (269; 54).

En raison de la complexité croissante et de l'interconnexion des systèmes automobiles, les méthodes de sécurité traditionnelles ne suffisent plus (94; 270). Les techniques traditionnelles, principalement axées sur la protection périmétrique, ont du mal à suivre les tactiques changeantes des adversaires cybernétiques. Cela souligne l'importance d'une approche plus approfondie et rigoureuse de la sécurité des systèmes automobiles. Les méthodes formelles et les approches de validation offrent une solution intéressante à ce problème croissant (132; 186; 124). Les approches formelles fournissent un cadre structuré pour exprimer et vérifier les propriétés de sécurité en utilisant une rigueur mathématique (279; 186). Les techniques de validation, telles que les évaluations pratiques et les scénarios d'attaque simulés, complètent les méthodes formelles en validant les mesures de sécurité dans le monde réel (134; 141).

L'objectif principal de cette étude de recherche est de présenter un aperçu complet des méthodologies formelles les plus récentes et des approches de validation utilisées dans le secteur automobile pour garantir la sécurité du système. Cela comprend une discussion approfondie des techniques de validation telles que les tests de pénétration et l'injection de fautes, ainsi qu'une exploration des différentes méthodes formelles allant de la vérification de modèles à la démonstration de théorèmes (167; 123).

Ce document couvre l'ensemble du processus de développement des systèmes automobiles, de l'ingénierie des exigences aux tests. Nous espérons mettre en évidence les points d'intégration où les méthodes formelles et les techniques de validation jouent un rôle essentiel dans le renforcement de la sécurité du système en examinant chaque phase du cycle de développement. Cette étude tient également compte des évolutions de la convergence des méthodes formelles, de l'apprentissage automatique et de la technologie de la blockchain, offrant un aperçu de l'avenir de la sécurité des systèmes automobiles.

Cette étude se veut une ressource importante pour les chercheurs, les praticiens et les décideurs intéressés par la conception, le développement et l'évaluation de systèmes automobiles sécurisés, en réalisant une enquête exhaustive de la littérature pertinente et en effectuant une analyse approfondie.

**Enquêtes connexes :** Le travail décrit dans (152) fournit une analyse approfondie des solutions existantes et des principaux défis liés à la sécurisation des systèmes de transport intelligents (STI). Il aborde les problèmes de sécurité associés à la connectivité croissante et à l'autonomie des véhicules dans les STI. Le travail comprend un tutoriel sur les problèmes de sécurité et les attaques, évalue les solutions existantes et discute des tendances récentes visant à renforcer la sécurité des STI. Dans l'ensemble, il vise à améliorer la sécurité et la confidentialité des STI en identifiant les menaces potentielles et en mettant en évidence les domaines à améliorer pour sécuriser ces systèmes.

Les auteurs de l'article (221) proposent un aperçu et une comparaison des protocoles et des schémas de communication pour les architectures automobiles basées sur le paradigme de l'architecture orientée services (SOA), par opposition aux approches traditionnelles orientées signaux. L'article met en évidence la nécessité d'une conception architecturale transformée pour répondre aux nouvelles exigences des clients et des fabricants, qui consistent notamment à ajouter de nouvelles fonctions logicielles au cours du cycle de vie du produit. Il aborde les défis et les opportunités des SOA en termes de sécurité de l'information et explore diverses contre-mesures de sécurité telles que les pare-feu, les systèmes de détection d'intrusion (IDS) et la gestion des identités et des accès (IAM) dans le contexte automobile. Le travail se conclut par une discussion sur l'adaptation des mesures de sécurité existantes et de leurs caractéristiques spécifiques dans une architecture hybride combinant à la fois des approches orientées signaux et orientées services.

La principale contribution de l'article (164) réside dans la fourniture d'une vue d'ensemble complète de l'analyse des menaces et de l'évaluation des risques (TARA) telle qu'elle concerne la cybersécurité des véhicules connectés dans l'industrie automobile. L'article met l'accent sur le potentiel d'accidents résultant de vulnérabilités en matière de cybersécurité compte tenu de l'interdépendance croissante avec les réseaux externes, soulignant ainsi l'importance de la cybersécurité. Le texte précise que les fabricants automobiles augmentent leurs investissements dans le développement de mécanismes de défense en matière de cybersécurité et propose TARA comme une approche efficace pour réduire les coûts et garantir une défense efficace lors des phases initiales de développement des véhicules. Une classification de diverses méthodologies TARA est proposée, et les approches existantes sont comparées. Une évaluation de l'efficacité des outils couramment utilisés pour TARA est également réalisée. En outre, l'article présente la notion de mise en correspondance attaque-défense, une approche qui vise à synchroniser les vulnérabilités et les menaces détectées avec les contre-mesures les plus appropriées. L'article se conclut par une discussion sur les développements futurs potentiels de TARA dans l'industrie automobile.

L'étude (84) contribue à une analyse approfondie des mises à jour logicielles sans fil (OTA) dans l'industrie automobile, en mettant particulièrement l'accent sur les problèmes de sécurité. Le rapport souligne l'importance croissante des mises à jour OTA pour les voitures connectées et leur utilisation pour améliorer à distance les fonctionnalités et corriger les erreurs logicielles. L'article vise à accroître la sensibilisation dans ce domaine en offrant un examen approfondi des divers domaines de recherche et techniques dans les technologies de mise à jour OTA. Il propose une analyse compa-

rative des méthodologies actuelles, examine les approches de mise à jour OTA viables et sécurisées, et parle de la relation entre les technologies de voiture connectée et les fonctionnalités de mise à jour OTA. La satisfaction des clients, les caractéristiques d'utilisabilité et les fonctionnalités des voitures prenant en charge les mises à jour OTA sont également examinées dans l'enquête. Le travail offre des perspectives importantes pour le développement des mises à jour OTA dans les voitures en soulignant les orientations futures de la recherche, notamment dans le domaine de la sécurité.

Une vue d'ensemble approfondie de la cybersécurité dans le contexte des véhicules connectés et autonomes (VCA) est la contribution principale de l'article (249). L'étude reconnaît les avantages potentiels des VCA, notamment de meilleures options de mobilité, des prix plus bas pour les utilisateurs, un transport plus sûr et la création de nouveaux emplois. Elle attire également l'attention sur la menace croissante des attaques malveillantes à la sécurité des VCA à mesure que l'automatisation et la connectivité augmentent. Sur la base des réseaux de communication et des objets d'attaque, l'enquête divise les menaces et les vulnérabilités en matière de cybersécurité en trois catégories : les attaques véhicule-vers-tout, les attaques dans le véhicule et les autres attaques. Elle aborde également les tactiques de défense pour protéger les VCA et parle du risque cybernétique comme type d'attaque dans l'environnement des VCA. Les exigences de cybersécurité et de sécurité disponibles pour les VCA sont présentées dans la conclusion de l'article, accompagnées de quelques conseils utiles. Enfin, elle met en évidence les problèmes et les questions non résolues qui nécessitent des études supplémentaires dans le domaine de la cybersécurité des VCA.

La construction d'une taxonomie exhaustive des attaques dans le domaine de l'automobile est la contribution principale de l'étude (204). L'article met l'accent sur l'évolution de l'industrie automobile au fil du temps et sur la manière dont l'ajout de composants électroniques aux voitures a entraîné de nouveaux vecteurs d'attaque et vulnérabilités. Une zone de recherche et de pratique qui fait défaut est une taxonomie d'attaque complète pour l'industrie automobile. Afin de résoudre ce problème, les auteurs réalisent une revue approfondie de la littérature, trouvent et catégorisent 48 attaques distinctes en utilisant la taxonomie suggérée des méthodes d'attaque. En reliant plusieurs vecteurs d'attaque, la taxonomie peut aider les testeurs de pénétration dans l'industrie automobile à créer des attaques plus complexes. Il s'agit d'un outil utile pour les testeurs de pénétration. Cinq dimensions - couches AUTOSAR, domaines d'attaque, principes de sécurité de l'information, surfaces d'attaque et profils d'attaquants - sont également utilisées pour catégoriser les vecteurs d'attaque découverts. Les résultats attirent l'attention sur les vecteurs d'attaque les plus couramment utilisés dans la littérature, qui sont principalement dirigés contre les couches d'application et de service de l'architecture AUTOSAR. Ces vecteurs comprennent le leurre GPS, l'injection de messages, l'usurpation de nœud, le sybil et les attaques de ver.

L'étude présentée dans (92) contribue à une évaluation approfondie de la gestion de la confiance dans l'écosystème de l'Internet des véhicules (IoV). L'article reconnaît l'IoV comme un moyen d'améliorer l'expérience utilisateur en fournissant des services avancés axés sur le confort et la sécurité. Les auteurs attirent l'attention sur le fait que l'architecture de l'Internet des véhicules (IoV) est complexe, impliquant des êtres humains, des objets routiers, des voitures et des systèmes de transport intelligents (ITS).

Ils soulignent que les différentes formes de communication au sein de l'Internet des véhicules (IoV) créent de nouveaux besoins en matière de sécurité et augmentent la surface d'attaque de l'écosystème. L'étude met l'accent sur l'utilisation de la gestion de la confiance comme technique de sécurité pour améliorer la fiabilité dans l'environnement de l'Internet des véhicules (IoV) afin de surmonter ces problèmes. Bien que la gestion de la confiance dans le contexte des réseaux ad hoc véhiculaires (VANET) ait été largement explorée, les auteurs soutiennent que les techniques créées pour les VANET doivent être modifiées et étendues pour être utiles dans le contexte plus large de l'Internet des véhicules. Ils attirent l'attention sur la manière dont des technologies de pointe telles que l'intelligence artificielle (38; 278; 184; 12; 32; 181; 130; 228; 128), l'informatique en nuage, la blockchain et les réseaux définis par logiciel (SDN) peuvent offrir des stratégies de gestion de la confiance plus appropriées et pratiques dans le contexte de l'Internet des véhicules.

La principale contribution de l'étude (116) est une revue systématique et une analyse des études précédemment menées sur les défenses et les attaques des véhicules autonomes. Les auteurs soulignent l'importance du transport intelligent au sein des villes intelligentes et les vulnérabilités potentielles des véhicules autonomes qui peuvent avoir des répercussions sur la vie humaine et la sécurité. Grâce à un examen approfondi de 151 articles publiés de 2008 à 2019, l'enquête examine en profondeur les attaques autonomes et les mécanismes de défense. Le texte établit trois catégories distinctes pour les attaques et les mécanismes de défense, en mettant l'accent sur l'importance de l'apprentissage automatique et de l'intelligence artificielle dans l'identification des anomalies. L'enquête fournit des constatations significatives et des implications pour les recherches ultérieures, en mettant particulièrement l'accent sur l'application de l'intelligence artificielle pour aborder les problèmes de sécurité associés aux véhicules autonomes dans les environnements de villes intelligentes.

Le travail présenté dans (207) est une étude approfondie qui examine les attaques actuelles et les techniques de défense utilisées pour les véhicules connectés et autonomes (VCA). Les auteurs soulignent l'influence significative des véhicules autonomes sur les systèmes de transport intelligents et les avantages potentiels des VCA pour un transport sûr et efficace. Néanmoins, il est important de souligner les obstacles de sécurité considérables auxquels les véhicules connectés et autonomes (VCA) sont confrontés en raison de leur nature interconnectée et de la susceptibilité de leurs composants aux attaques potentielles. L'étude examine 189 articles publiés entre 2000 et 2020, en se concentrant spécifiquement sur 131 articles qui abordent les modèles d'attaque ou les stratégies de défense pour les véhicules connectés et autonomes (VCA). Cette étude offre un examen complet des attaques de sécurité et des contre-mesures pour les véhicules connectés et autonomes (VCA). Elle explore différents modèles d'attaque basés sur les composants ciblés, les exigences d'accès et les motivations. De plus, elle identifie les défis de recherche actuels et les tendances dans ce domaine. L'étude révèle un écart entre la recherche académique et la mise en œuvre industrielle des problèmes de sécurité liés aux véhicules connectés et autonomes (VCA). Cela souligne l'importance d'améliorer les défenses dans le développement futur des VCA. L'étude améliore la compréhension de l'état actuel et des tendances en matière de sécurité des VCA, offrant des informations précieuses aux chercheurs et aux ingénieurs souhaitant améliorer

la sécurité des VCA.

L'étude (177) présente une étude approfondie qui examine les problèmes de sécurité et de confidentialité associés à l'informatique en nuage véhiculaire (VCC). Les avantages potentiels de la VCC dans la transformation des services informatiques pour le transport intelligent, la conduite autonome et d'autres applications diverses sont dûment reconnus par les auteurs. Néanmoins, ils reconnaissent que la VCC est limitée par d'importants problèmes de confidentialité et de sécurité. L'article présente un examen à jour de l'architecture, des fonctionnalités et des mises en œuvre de la VCC. L'article réalise une enquête exhaustive sur les problèmes de sécurité et de confidentialité dans la VCC, en couvrant différentes couches, notamment la couche de ressources physiques, la couche de réseau véhicule-vers-tout (V2X), la couche de cloud véhiculaire et l'ensemble du système. Il propose également une taxonomie pour l'identification des menaces. La recherche met l'accent sur les domaines de préoccupation et les questions non résolues, fournissant des informations significatives pouvant orienter les futures recherches sur les défis de sécurité et de confidentialité associés à la VCC. Dans son ensemble, cette étude contribue au corpus de connaissances en fournissant une compréhension approfondie des problèmes de sécurité et de confidentialité propres à la VCC. Cela permet à son tour de formuler des mesures efficaces pour renforcer la sécurité et la confidentialité des systèmes de VCC.

L'enquête présentée dans cet article se distingue des enquêtes existantes en fournissant un aperçu complet des méthodes formelles et des techniques de validation pour la sécurité des systèmes automobiles, couvrant un large éventail d'approches et leur intégration dans le cycle de développement. Contrairement à d'autres enquêtes qui se concentrent sur des aspects spécifiques tels que les protocoles, les menaces ou les attaques, notre article adopte une approche holistique, examinant l'ensemble du spectre des méthodes formelles et des techniques de validation, leur application dans différentes phases de développement, et en identifiant les tendances et les défis de recherche actuels. Cette perspective globale fait de notre enquête une ressource précieuse pour les chercheurs, les praticiens et les décideurs impliqués dans la conception et le développement de systèmes automobiles sécurisés.

## 2 Défis de la sécurité des systèmes automobiles

Le domaine des systèmes automobiles rencontre une multitude d'obstacles dans le domaine de l'assurance de la sécurité. La complexité croissante et l'interconnectivité de ces systèmes ont donné naissance à de nouvelles vulnérabilités et risques susceptibles de provoquer des violations de sécurité (27; 225). Comprendre et atténuer efficacement ces problèmes est d'une importance capitale pour garantir la protection des systèmes automobiles contre les atteintes potentielles à la sécurité (245; 185). Cette section se penche sur les principaux obstacles liés à la sécurité au sein des systèmes automobiles.

## 2.1 Types de réseaux et protocoles de communication dans les systèmes automobiles

Les systèmes automobiles reposent sur divers réseaux et protocoles de communication pour permettre l'échange d'informations entre les différents composants et sous-systèmes. Comprendre les types de réseaux et les protocoles de communication utilisés dans les systèmes automobiles est crucial pour assurer une communication efficace et sécurisée à l'intérieur du véhicule. Voici quelques types de réseaux couramment utilisés dans les systèmes automobiles :

- **Controller Area Network (CAN) :** CAN est un réseau largement utilisé dans les véhicules qui facilite la communication entre les unités de contrôle électronique (ECU). Il a été initialement développé dans les années 1980 et est depuis devenu la norme de facto pour la communication embarquée. CAN est un réseau robuste, peu coûteux et tolérant aux erreurs qui prend en charge les applications en temps réel. Il fonctionne selon une topologie en bus, où plusieurs ECU sont connectées à un bus de communication partagé. CAN permet une communication fiable et efficace entre différents systèmes du véhicule, tels que l'unité de commande du moteur, l'unité de commande de la transmission et le module de contrôle de la carrosserie.
- **Local Interconnect Network (LIN) :** LIN est un autre réseau couramment utilisé dans les systèmes automobiles, principalement pour la communication entre les composants moins critiques. Il offre une solution économique pour les besoins de communication à basse vitesse, tels que le contrôle des interrupteurs de vitre, des verrous de porte et de l'éclairage intérieur. LIN fonctionne selon une architecture maître-esclave, où un nœud maître communique avec plusieurs nœuds esclaves. Comparé à CAN, LIN a une bande passante plus faible et est conçu pour des applications plus simples et moins critiques en termes de temps.
- **Ethernet :** Ethernet est de plus en plus adopté dans les véhicules modernes en raison de ses capacités à haut débit. Il permet la communication entre divers ECU et prend en charge des applications avancées telles que les systèmes d'infodivertissement, les systèmes d'aide à la conduite avancés (ADAS) et la conduite autonome. L'Ethernet automobile est basé sur la norme Ethernet, mais comprend des fonctionnalités supplémentaires pour répondre aux exigences spécifiques des applications automobiles. Il offre des débits de données plus élevés, une fiabilité améliorée et la possibilité de hiérarchiser différents types de trafic.
- **FlexRay :** FlexRay est un réseau déterministe et tolérant aux erreurs qui assure une communication haut débit dans les systèmes automobiles critiques pour la sécurité. Il a été développé pour répondre aux exigences strictes des systèmes avancés d'aide à la conduite et des applications x-by-wire. FlexRay prend en charge à la fois la communication déclenchée par le temps et la communication déclenchée par des événements, permettant une transmission précise et prévisible des données. Il offre une bande passante élevée, une tolérance aux erreurs et des capacités de synchronisation, ce qui le rend adapté aux applications critiques nécessitant une communication en temps réel.
- **Media-Oriented Systems Transport (MOST) :** MOST est une technologie de réseau principalement utilisée dans les systèmes multimédias et d'infodiver-

tissement automobiles. Elle permet la transmission de l'audio, de la vidéo et des données de contrôle entre différents appareils multimédias dans le véhicule, tels que les unités principales, les amplificateurs et les écrans. MOST prend en charge le transfert de données à haute vitesse et offre des fonctionnalités telles que le streaming synchrone, la gestion du réseau et la tolérance aux erreurs.

— **Ethernet automobile :** L'Ethernet automobile est une extension de la norme Ethernet spécifiquement conçue pour les applications automobiles. Il permet une communication à haut débit et répond aux exigences croissantes en matière de bande passante des véhicules modernes. L'Ethernet automobile permet l'intégration de différents systèmes tels que l'infodivertissement, les ADAS et la diagnostic du véhicule sur une infrastructure réseau unique. Il utilise des protocoles et des technologies Ethernet, tels que Ethernet AVB (audio video bridging) et TSN (time-sensitive networking), pour garantir une communication fiable et déterministe.

— **Réseaux sans fil :** Les réseaux sans fil jouent un rôle essentiel dans les systèmes automobiles, permettant la connectivité avec des appareils et des services externes. Par exemple, le Bluetooth est couramment utilisé pour les appels mains libres, la diffusion audio et la connectivité sans fil des appareils. Le Wi-Fi peut fournir un accès Internet embarqué, permettant aux passagers de connecter leurs appareils et d'accéder aux services en ligne. Les réseaux cellulaires, tels que la 4G LTE et la 5G, permettent la communication entre véhicules (V2V) et entre véhicules et infrastructure (V2I), prenant en charge des fonctionnalités telles que les diagnostics à distance, les mises à jour over-the-air et les services connectés.

— **CAN-FD :** CAN avec débit de données flexible (CAN-FD) est une extension du protocole CAN traditionnel qui permet des débits de données plus élevés et des tailles de charge utile accrues. Il répond à la demande croissante de bande passante dans les systèmes automobiles, en particulier pour les applications nécessitant une transmission de données plus importante, telles que les données de capteurs haute résolution provenant de caméras et de radars. CAN-FD est rétrocompatible avec les réseaux CAN existants, permettant une transition en douceur vers des débits de données plus élevés.

— **LIN Sub-bus :** Le LIN Sub-bus est une extension du protocole LIN qui permet l'expansion des réseaux LIN pour accueillir plus d'appareils. Il permet la connexion de nœuds esclaves supplémentaires à un bus LIN existant, augmentant ainsi la capacité globale et la flexibilité du réseau. Le LIN Sub-bus est couramment utilisé dans les systèmes automobiles où le nombre de composants dépasse la capacité d'un seul bus LIN, comme dans les modules de porte complexes ou les groupes d'instruments.

Ces réseaux et protocoles de communication contribuent au paysage de communication diversifié des systèmes automobiles. La sélection et l'intégration de ces réseaux dépendent de facteurs tels que l'architecture spécifique du véhicule, les exigences de communication et les fonctionnalités souhaitées du véhicule. En utilisant efficacement ces réseaux et protocoles, les systèmes automobiles peuvent atteindre une communication efficace et fiable, permettant le fonctionnement transparent de divers sous-systèmes et améliorant l'expérience de conduite globale.

## 2.2 Défis de sécurité

Les systèmes de véhicules contemporains dépendent principalement des logiciels et de la connectivité réseau, les exposant ainsi à de multiples vulnérabilités. La présence de ces vulnérabilités peut provenir de divers facteurs, notamment des déficiences de conception, des erreurs d'implémentation, des protocoles de sécurité insuffisants ou de l'utilisation de composants logiciels obsolètes. La résolution de ces défis de sécurité est cruciale pour se protéger contre les menaces cybernétiques potentielles et garantir la sécurité et la confidentialité des occupants du véhicule. Voici quelques-uns des principaux défis de sécurité associés aux protocoles de réseau automobile :

- **Canaux de communication non sécurisés** (115; 273) : Les procédures de cryptage ou d'authentification insuffisantes mises en œuvre dans les protocoles de communication peuvent compromettre la confidentialité des données sensibles et faciliter l'accès illégal aux fonctionnalités du véhicule. Les attaquants peuvent intercepter ou manipuler les messages de communication, ce qui peut entraîner un accès non autorisé ou un contrôle non autorisé des systèmes critiques du véhicule.
- **Authentification et autorisation faibles** (111; 256) : Les mesures d'authentification et d'autorisation insuffisantes ou mal mises en œuvre peuvent accorder un accès non autorisé aux systèmes vitaux du véhicule, permettant aux attaquants de prendre le contrôle sans autorité appropriée. Les mécanismes d'authentification faibles peuvent permettre à des individus non autorisés de contourner les barrières de sécurité et d'accéder de manière non autorisée aux fonctions du véhicule ou aux données sensibles.
- **Vulnérabilités logicielles** (88; 188) : L'exploitation des vulnérabilités dans les composants logiciels tels que les systèmes d'exploitation, les systèmes d'infodivertissement ou le micrologiciel de la voiture peut influencer ou interrompre le fonctionnement des véhicules. Les vulnérabilités logicielles peuvent être exploitées par des attaquants pour prendre le contrôle des systèmes critiques, perturber les opérations du véhicule ou compromettre la sécurité des occupants du véhicule.
- **Pratiques de codage sécurisé inadéquates** (162; 75; 180) : L'absence de conformité aux principes de codage sécurisé dans le processus de développement logiciel peut créer plusieurs vulnérabilités, notamment les débordements de tampon, les injections SQL et les attaques par injection de code. Des pratiques de codage sécurisé inadéquates augmentent le risque de vulnérabilités logicielles, qui peuvent être exploitées par des attaquants pour accéder de manière non autorisée ou exécuter un code malveillant dans les systèmes du véhicule.
- **Exploitation de l'accès physique** (22; 57) : La sécurité de l'ensemble du système peut être compromise par des individus qui ont un accès physique au véhicule et exploitent les failles des interfaces de diagnostic, des systèmes embarqués ou des dispositifs antieffraction. Les attaquants ayant un accès physique peuvent manipuler ou altérer les composants du véhicule, compromettant l'intégrité et le fonctionnement des systèmes critiques.
- **Sécurité du bus CAN** : L'utilisation répandue du réseau de zone de commande (CAN) dans les systèmes automobiles en fait une cible attrayante pour

les attaques cybernétiques. Le CAN ne dispose pas de fonctionnalités de sécurité intégrées, le rendant vulnérable à diverses menaces, telles que le détournement de messages, les attaques de rejeu et l'accès non autorisé. Les attaquants peuvent manipuler les messages CAN pour compromettre desles fonctions critiques du véhicule, telles que le freinage ou la direction. La sécurisation du bus CAN nécessite la mise en place de mécanismes d'authentification, de cryptage et de détection des intrusions pour prévenir l'accès non autorisé et garantir l'intégrité et la confidentialité des communications.

— **Sécurité Ethernet** : L'Ethernet est de plus en plus utilisé dans les véhicules, notamment pour les applications avancées telles que l'infodivertissement et les systèmes d'aide à la conduite (ADAS). Cependant, les réseaux Ethernet sont confrontés à des défis de sécurité similaires à ceux des réseaux informatiques traditionnels. Cela comprend le risque d'accès non autorisé, de falsification de données et d'attaques par déni de service. La sécurisation de l'Ethernet dans les systèmes automobiles implique la mise en place de mécanismes robustes de contrôle d'accès, de protocoles de cryptage et de segmentation du réseau pour isoler les systèmes critiques des systèmes non critiques.

— **Sécurité des réseaux sans fil** : Les réseaux sans fil, tels que Bluetooth, Wi-Fi et les réseaux cellulaires, sont exposés à diverses menaces de sécurité. Par exemple, les connexions Bluetooth peuvent être vulnérables à l'écoute indiscrète et à l'appariement de périphériques non autorisés. Les réseaux Wi-Fi des véhicules peuvent être ciblés par des attaquants cherchant à accéder de manière non autorisée aux systèmes du véhicule ou à voler des données sensibles. Les réseaux cellulaires peuvent être exploités pour des attaques à distance, telles que la compromission du système de télématique ou des systèmes d'infodivertissement du véhicule. La sécurisation des réseaux sans fil nécessite la mise en œuvre de mécanismes d'authentification, de cryptage et de détection des intrusions solides pour se protéger contre l'accès non autorisé et les violations de données.

— **Sécurité FlexRay et LIN** : Bien que FlexRay et LIN soient moins souvent ciblés par des attaquants externes en raison de leur connectivité externe limitée, ils font toujours face à des défis de sécurité. Ces protocoles peuvent être vulnérables à des attaques physiques, telles que la manipulation des câbles de communication ou l'injection de signaux malveillants. La sécurisation des réseaux FlexRay et LIN implique la mise en place de mesures de sécurité physiques, telles que des câblages résistants aux intrusions et des connecteurs sécurisés, pour prévenir l'accès non autorisé et la manipulation.

— **Mises à jour logicielles sécurisées** : De nombreux systèmes automobiles dépendent de mises à jour logicielles pour corriger les vulnérabilités et introduire de nouvelles fonctionnalités. Cependant, le processus de mise à jour logicielle via le réseau présente des risques de sécurité. Les attaquants peuvent tenter d'exploiter les vulnérabilités du processus de mise à jour pour injecter un code malveillant ou altérer le logiciel. Les mécanismes de mise à jour logicielle sécurisée, tels que la signature de code, le démarrage sécurisé et les protocoles de mise à jour sécurisée, sont essentiels pour garantir l'intégrité et l'authenticité des mises à jour logicielles et prévenir les modifications non autorisées.

La résolution de ces défis de sécurité nécessite une approche globale qui englobe la conception de réseaux sécurisés (*secure network design*), le cryptage robuste (*robust encryption*), les mécanismes d'authentification solides (*strong authentication mechanisms*), les bonnes pratiques de développement logiciel sécurisé (*secure software development practices*), les mesures de sécurité physique (*physical security measures*), et les mécanismes de mise à jour logicielle sécurisée (*secure software update mechanisms*). Des évaluations de sécurité régulières (*regular security assessments*), des tests de vulnérabilité (*vulnerability testing*), et une surveillance continue (*continuous monitoring*) sont essentiels pour identifier et atténuer les risques de sécurité potentiels dans les protocoles de réseau automobile. En abordant ces défis, les systèmes automobiles peuvent améliorer la sécurité de leurs réseaux de communication, se protéger contre les menaces cybernétiques, et garantir la sécurité et la confidentialité des occupants du véhicule.

## 2.3 Autres risques possibles

L'exploitation de ces vulnérabilités peut entraîner des conséquences importantes, telles que la manipulation illégale des fonctions du véhicule, la compromission de systèmes cruciaux et la mise en danger de la sécurité à la fois des occupants et des autres personnes sur la route. Des exemples illustratifs de conséquences probables comprennent :

— Contrôle à distance non autorisé (248; 80) : L'acquisition non autorisée du contrôle de diverses opérations du véhicule, telles que la direction, le freinage et l'accélération, par des individus malveillants peut entraîner des risques importants pour la sécurité des occupants du véhicule et des autres personnes sur la route.
— Violations de la confidentialité des données (199; 222) : Les failles de sécurité dans les systèmes automobiles peuvent conduire à l'acquisition, au vol ou à la modification illégale d'informations confidentielles, telles que des données personnelles, des données de localisation ou des modèles d'utilisation du véhicule.
— Injection de logiciels malveillants (250; 214) : L'injection de logiciels malveillants par des acteurs malveillants dans les systèmes automobiles a le potentiel de perturber les opérations, de compromettre les fonctions critiques pour la sécurité ou de faciliter l'observation et le suivi illégaux.
— Risques pour la sécurité physique (37; 277) : Les failles de sécurité ont le potentiel de donner lieu à des risques pour la sécurité physique, tels que la désactivation des dispositifs de sécurité, la manipulation du déploiement des airbags ou l'interférence avec le système de freinage antiblocage (ABS), ce qui peut entraîner des accidents ou des blessures.

## 2.4 Limitations des techniques classiques

Les exemples énumérés ci-dessus soulignent l'importance de résoudre les vulnérabilités et de garantir la sécurité des systèmes automobiles afin de prévenir de manière proactive les accidents potentiels et de réduire efficacement les risques associés. Bien que des techniques classiques aient été utilisées dans le domaine de la sécurité des systèmes automobiles, elles présentent certaines limites qui les rendent moins efficaces

pour faire face aux menaces et aux vulnérabilités en constante évolution. Dans cette section, nous discuterons de certaines des techniques classiques couramment utilisées dans l'industrie automobile et de leurs principales limitations :

— Firewalls et systèmes de détection d'intrusion (IDS) (73; 156; 229; 208) : Les pare-feu établissent une barrière entre les réseaux internes et externes, tandis que les IDS surveillent le trafic réseau à la recherche d'activités suspectes. Cependant, ces techniques présentent des limites dans le contexte des systèmes automobiles. Elles peuvent ne pas être en mesure de détecter des attaques exploitant des vulnérabilités au sein du réseau interne du véhicule, telles que des composants compromis qui communiquent entre eux. De plus, elles ont souvent du mal à suivre la complexité et la sophistication croissantes des attaques, car les attaquants trouvent continuellement de nouvelles façons de contourner ou de contourner les mesures de sécurité réseau traditionnelles.
— Cryptage et protocoles de communication sécurisés (66; 112; 59; 133) : Le cryptage et les protocoles de communication sécurisés sont essentiels pour protéger les données sensibles transmises entre les différents composants d'un système automobile. Ces techniques garantissent la confidentialité et l'intégrité de la communication. Cependant, le cryptage seul ne peut pas empêcher les attaques exploitant d'autres vulnérabilités du système. De plus, les mécanismes de gestion et de distribution des clés dans les systèmes automobiles peuvent être difficiles à mettre en œuvre de manière sécurisée, et si ces mécanismes sont compromis, l'efficacité du cryptage peut être gravement compromise.
— Contrôle d'accès et mécanismes d'authentification (11; 194; 61; 24) : Le contrôle d'accès et les mécanismes d'authentification sont utilisés pour restreindre l'accès aux fonctions et aux ressources critiques d'un système automobile. Ces techniques aident à prévenir le contrôle et la manipulation non autorisés du véhicule. Cependant, elles reposent sur l'hypothèse que les mécanismes d'authentification eux-mêmes sont sécurisés. Des mécanismes d'authentification faibles ou mal mis en œuvre peuvent être exploités par des attaquants pour obtenir un accès non autorisé. De plus, dans des systèmes automobiles complexes avec de nombreux composants interconnectés, la gestion et l'application des politiques de contrôle d'accès peuvent devenir de plus en plus difficiles.
— Pratiques de développement logiciel sécurisé (191; 252; 14; 239) : Les pratiques de développement logiciel sécurisé, telles que les directives de codage sécurisé et l'analyse de vulnérabilités, sont cruciales pour la construction de systèmes automobiles robustes et résilients. Ces pratiques visent à éliminer les vulnérabilités logicielles courantes et à réduire la surface d'attaque. Cependant, elles ne peuvent garantir l'absence de toutes les vulnérabilités, en particulier dans des systèmes complexes avec de nombreux composants logiciels et interactions. De plus, l'incorporation de pratiques de développement logiciel sécurisé nécessite des efforts et une expertise significatifs, et il peut être difficile de les appliquer de manièrecohérente à toutes les étapes du processus de développement.
— Mesures de sécurité physique (237; 236; 233; 263) : Des mesures de sécurité physique, telles que des mécanismes inviolables et des interfaces de diagnostic sécurisées, sont utilisées pour protéger les systèmes automobiles contre les at-

taques physiques. Bien que ces mesures soient importantes, elles peuvent ne pas être suffisantes pour se défendre contre des attaquants sophistiqués ayant un accès physique au véhicule. Des attaquants déterminés peuvent contourner ou manipuler les mesures de sécurité physique avec suffisamment de temps et de ressources. De plus, les mesures de sécurité physique ne peuvent pas remédier aux vulnérabilités découlant d'attaques logicielles ou réseau, qui sont de plus en plus courantes dans les systèmes automobiles modernes.

Il est important de reconnaître les limites des techniques classiques et d'explorer des approches plus avancées et plus complètes en matière de sécurité des systèmes automobiles. Dans les sections suivantes, nous examinerons l'utilisation des méthodes formelles, des techniques de validation et d'autres stratégies émergentes pour pallier ces limitations et améliorer la sécurité des systèmes automobiles.

# 3 Méthodes formelles pour l'analyse de la sécurité des systèmes automobiles

Afin de garantir la sécurité des systèmes automobiles, les méthodes formelles se sont développées comme des outils puissants pour l'analyse et la vérification de leurs propriétés de sécurité (137; 138; 126). Cette section examine de nombreuses méthodologies formelles notables utilisées dans le domaine de la sécurité des systèmes automobiles (212; 72; 55). Elles permettent aux concepteurs de systèmes de spécifier le comportement et les propriétés du système à l'aide de notations mathématiques précises telles que des formules logiques et des machines à états (140; 145; 168; 147; 132; 129). Les techniques formelles peuvent ensuite utiliser des outils automatisés pour examiner le comportement et les attributs du système, tels que la cohérence, l'exhaustivité et la correction (167; 121; 166).

## 3.1 Vérification de modèles

La vérification de modèles est une technique rigoureuse qui consiste à examiner systématiquement un modèle à états finis d'un système afin de déterminer s'il est conforme à une propriété ou une spécification donnée (28; 154; 43; 42; 169; 148; 122; 144; 123; 149; 136; 170; 139; 141). Dans le domaine de la sécurité des systèmes automobiles, l'application de la vérification de modèles peut s'avérer une approche précieuse pour examiner les éléments liés à la sécurité d'un système donné (266; 176; 21; 192; 124; 143; 150; 125; 31; 127).

Formellement, soit $M = (S, I, T, AP, L)$ un modèle à états finis, où :

— $S$ est un ensemble fini d'états représentant les configurations possibles du système.
— $I \subseteq S$ est l'ensemble des états initiaux à partir desquels l'exécution du système débute.
— $T \subseteq S \times S$ est la relation de transition qui spécifie les transitions d'états possibles du système.
— $AP$ est un ensemble de propositions atomiques représentant les propriétés d'intérêt qui peuvent être satisfaites dans un état.

— $L : S \to 2^{AP}$ est une fonction d'étiquetage qui associe à chaque état l'ensemble des propositions atomiques satisfaites dans cet état.

Le processus de vérification de modèles consiste à vérifier si une propriété donnée $\varphi$ est satisfaite dans le modèle $M$. La propriété $\varphi$ est exprimée à l'aide d'un formalisme de logique temporelle, tel que la logique temporelle linéaire (LTL) ou la logique des arbres de calcul (CTL). La vérification est effectuée en explorant exhaustivement l'espace des états du modèle et en vérifiant si la propriété est satisfaite dans chaque état et transition.

Par exemple, considérons un système automobile qui intègre un protocole cryptographique pour assurer une communication sécurisée entre différents composants. Soit $M = (S, I, T, AP, L)$ le modèle à états finis du système, où $S$ représente les états possibles de l'exécution du protocole, $I$ désigne les états initiaux, $T$ spécifie les transitions entre les états, $AP$ inclut les propositions atomiques relatives aux propriétés de sécurité, et $L$ est la fonction d'étiquetage qui associe les états aux propositions atomiques satisfaites.

Pour vérifier une propriété de sécurité $\varphi$ dans le modèle $M$, nous utilisons une formule de logique temporelle, telle qu'une formule LTL. Par exemple, $\varphi$ peut exprimer la propriété "la confidentialité est satisfaite dans toutes les exécutions du protocole", ce qui peut être formalisé comme $\Box$(Confidentialité). Le processus de vérification de modèles explore alors systématiquement toutes les exécutions du protocole possibles, vérifiant si la formule $\varphi$ est satisfaite dans chaque état et transition. Si une violation est trouvée, le processus de vérification de modèles fournit un contre-exemple, tel qu'une séquence de messages de protocole qui conduit à une vulnérabilité de sécurité, comme une attaque de rejeu ou une compromission de clé.

Grâce à une exploration méthodique de tous les états possibles d'un système, le processus de vérification de modèles peut détecter efficacement les vulnérabilités, déterminer les voies potentielles d'attaques et valider l'intégrité des fonctionnalités de sécurité. Cette technique s'avère très avantageuse pour identifier les vulnérabilités de sécurité complexes qui peuvent survenir en raison d'interactions complexes à l'intérieur d'un système.

## 3.2 Démonstration de théorèmes

Le processus de démonstration de théorèmes est une méthodologie formelle basée sur les principes de la logique mathématique et de la théorie de la preuve afin de déterminer l'exactitude et la validité d'un système donné (107; 120; 90; 163). Il consiste à construire une preuve formelle qui établit la vérité d'une proposition ou vérifie la validité d'une affirmation sur la base d'un ensemble d'axiomes, de règles d'inférence et de raisonnements logiques (91; 78; 30; 44).

Formellement, soit $\Gamma$ un ensemble d'axiomes et $P$ une proposition ou une déclaration. Le processus de démonstration de théorèmes vise à démontrer que $P$ est vraie sur la base des axiomes et des déductions logiques qui en découlent. Cela est généralement réalisé en construisant une preuve formelle, qui est une séquence d'étapes logiques qui établissent la vérité de $P$ en utilisant des règles d'inférence valides. La preuve peut impliquer l'application de règles logiques, telles que le modus ponens ou la généralisa-

tion universelle, et des déductions logiques basées sur les axiomes et les énoncés déjà prouvés.

Dans le domaine de la sécurité des systèmes automobiles, la démonstration de théorèmes peut être appliquée pour examiner et analyser les propriétés de sécurité relatives aux différents composants du système (4; 215; 216; 158). Par exemple, elle peut contribuer à valider l'exactitude des méthodes de sécurité, telles que les procédures d'authentification ou les algorithmes de chiffrement, utilisées pour protéger les données critiques au sein du système automobile. En formalisant les attributs de sécurité sous forme de propositions et en utilisant des prouveurs de théorèmes automatisés ou interactifs, il devient possible de démontrer logiquement l'absence de certaines vulnérabilités ou de valider l'exactitude des systèmes de sécurité.

Par exemple, considérons un système automobile qui utilise un mécanisme d'authentification pour accorder l'accès à des fonctionnalités privilégiées. L'approche de démonstration de théorèmes peut être utilisée pour vérifier les propriétés de sécurité du protocole d'authentification. Les axiomes dans ce cas pourraient inclure les propriétés d'un schéma d'authentification sécurisé, telles que l'unicité des informations d'identification de l'utilisateur et la résistance aux attaques d'usurpation d'identité. En appliquant des règles logiques et en effectuant des déductions basées sur les axiomes, une preuve formelle peut être construite pour établir la validité du protocole d'authentification, garantissant ainsi sa conformité aux propriétés de sécurité souhaitées.

L'utilisation de la démonstration de théorèmes offre une technique méthodique et logique pour garantir la sécurité des systèmes automobiles. Elle permet la vérification formelle des propriétés de sécurité et offre une approche rigoureuse pour vérifier l'exactitude et la validité des mécanismes de sécurité.

### 3.3 Interprétation Abstraite

L'interprétation abstraite est une technique rigoureuse qui offre une approche structurée pour estimer le comportement d'un système en abstrayant ses composants concrets (244; 48; 63). Elle fournit un cadre d'analyse des programmes ou des systèmes en approximant leurs comportements à l'aide de représentations abstraites (46; 47; 219). Ces représentations abstraites capturent les propriétés essentielles du système en ignorant les détails non pertinents, ce qui permet une analyse scalable et facilite l'identification de vulnérabilités de sécurité (33; 23; 76).

Formellement, soit $C$ le domaine concret représentant les comportements réels d'un système, et soit $A$ le domaine abstrait représentant une approximation des comportements concrets. Le processus d'interprétation abstraite consiste à définir une paire de fonctions : une fonction d'abstraction sonore $\alpha : C \to A$, qui fait correspondre des éléments concrets à des éléments abstraits, et une fonction de concrétisation $\gamma : A \to C$, qui fait correspondre des éléments abstraits à des éléments concrets. Ces fonctions établissent une connexion formelle entre les domaines concret et abstrait, garantissant que les représentations abstraites préservent les propriétés essentielles des comportements concrets.

Dans le domaine de la sécurité des systèmes automobiles, l'interprétation abstraite peut être utilisée pour examiner et analyser les attributs de sécurité des composants logiciels (255; 210; 265). Par exemple, les techniques d'interprétation abstraite peuvent

être utilisées pour analyser les modules logiciels et identifier les vulnérabilités en abstrayant leur comportement et en analysant le comportement hypothétique d'un attaquant. En créant des modèles abstraits qui capturent les propriétés de sécurité essentielles et en simulant des attaques potentielles, il est possible d'identifier les faiblesses de sécurité et de développer les contre-mesures appropriées.

De plus, l'interprétation abstraite peut contribuer à l'évaluation de l'efficacité des mesures de sécurité et à l'évaluation de la capacité du système à résister aux attaques. En abstrayant les comportements du système, des propriétés de sécurité telles que la confidentialité ou l'intégrité peuvent être définies et vérifiées sur les représentations abstraites. Cela permet d'estimer les garanties de sécurité du système et offre des informations sur l'impact potentiel des attaques.

L'utilisation de l'interprétation abstraite offre une combinaison harmonieuse d'exactitude et de capacité à traiter des systèmes automobiles étendus, ce qui en fait une option pragmatique pour des fins d'analyse. Elle permet une analyse scalable en abstrayant les comportements du système et offre une approche systématique pour identifier les vulnérabilités de sécurité et évaluer l'efficacité des mesures de sécurité.

Par exemple, considérons un système automobile qui intègre un protocole de communication entre différents composants. L'interprétation abstraite peut être appliquée pour abstraire le comportement du protocole et analyser ses propriétés de sécurité. Les comportements concrets du protocole, tels que les échanges de messages et les opérations de chiffrement, peuvent être abstraits vers une représentation de plus haut niveau. Les comportements hypothétiques d'un attaquant peuvent également être abstraits et analysés pour identifier les vulnérabilités potentielles, telles que le détournement de messages ou les attaques de rejeu. En raisonnant sur les représentations abstraites, il est possible de détecter les faiblesses de sécurité et de concevoir les mesures de sécurité appropriées.

## 3.4 Autres Méthodes Formelles Pertinentes

Outre les techniques de vérification de modèles, de démonstration de théorèmes et d'interprétation abstraite, il existe diverses autres méthodes formelles pertinentes pour l'analyse de la sécurité des systèmes automobiles :

— **Analyse Statique** (25; 182; 106) : Les approches d'analyse statique consistent à examiner le code d'un programme ou les spécifications d'un système sans les exécuter, dans le but d'identifier d'éventuelles vulnérabilités de sécurité. En analysant la structure du code, le flux de données et les dépendances au sein d'un système, les techniques d'analyse statique peuvent détecter efficacement des problèmes de sécurité courants, tels que les dépassements de tampon ou les procédures de manipulation incorrecte des données. En examinant la structure du code, l'analyse statique permet d'identifier les erreurs de codage potentielles, telles que les variables non initialisées ou la validation insuffisante des entrées, qui peuvent conduire à des vulnérabilités de sécurité. De plus, l'analyse statique permet de détecter les méthodes de manipulation de données non sécurisées, telles que le stockage non sécurisé d'informations sensibles ou les protections insuffisantes contre les fuites d'informations. L'analyse des flux de données et des dépendances permet à l'analyse statique de détecter les vulnérabilités de sécu-

rité potentielles pouvant résulter de l'interaction entre différents composants du système. Ces vulnérabilités incluent le traitement inadéquat des entrées utilisateur et la transmission de données non sécurisées entre différents modules. Les outils d'analyse statique utilisent fréquemment des algorithmes avancés pour examiner le code à grande échelle, ce qui les rend bien adaptés aux bases de code complexes de l'industrie automobile.

— **Exécution Symbolique** (117; 146; 6; 81; 19) : L'exécution symbolique est une méthode systématique qui consiste à examiner délibérément plusieurs chemins dans le code d'un programme tout en considérant des entrées symboliques. L'objectif de cette approche est d'identifier les vulnérabilités et de générer des cas de test. L'exécution symbolique est une technique qui permet d'explorer différents chemins d'exécution et de générer des entrées qui peuvent tester différents comportements du programme. Cela est réalisé en exécutant un programme de manière symbolique. Cette fonctionnalité permet d'identifier les chemins d'attaque possibles et peut aider à la création de scénarios de test ciblés. L'exécution symbolique est capable de détecter les entrées qui peuvent activer des vulnérabilités de sécurité, telles que des circonstances de chemin conduisant à des dépassements de tampon ou des entrées contournant les mesures d'authentification. L'exécution symbolique est une technique précieuse qui peut être utilisée pour générer des cas de test complets englobant différents comportements du programme, y compris des cas limites et des scénarios extraordinaires. Cette approche a le potentiel de révéler des problèmes de sécurité dissimulés. Cependant, l'utilisation de l'exécution symbolique peut rencontrer le problème de l'explosion de chemins lorsqu'elle est confrontée à des programmes complexes, ce qui présente des obstacles significatifs en termes de scalabilité. Différentes méthodologies, telles que la résolution de contraintes et la réduction de chemins, sont utilisées pour résoudre ces difficultés et améliorer la faisabilité de l'exécution symbolique dans l'analyse des systèmes automobiles.

— **Analyse des Protocoles de Sécurité** (240; 272; 56; 153; 52; 49) : L'objectif principal de l'analyse des protocoles de sécurité est d'évaluer les protocoles cryptographques utilisés dans les systèmes automobiles afin de garantir une communication sécurisée et la transmission des données. Cette méthodologie permet d'identifier les vulnérabilités potentielles dans les protocoles, telles que les attaques de rejeu ou les déficiences dans les systèmes d'échange de clés, contribuant ainsi à l'amélioration de la sécurité de la communication dans le système automobile. Le processus d'analyse des protocoles de sécurité implique la modélisation formelle des protocoles et leur soumission à des méthodologies d'analyse rigoureuses, telles que la vérification formelle ou l'analyse spécifique aux protocoles. Les approches de vérification académique, telles que la vérification de modèles ou la démonstration de théorèmes, peuvent être utilisées pour garantir l'exactitude des implémentations des protocoles et assurer leur conformité avec les caractéristiques de sécurité requises. Les techniques d'analyse spécifiques aux protocoles se concentrent principalement sur l'identification des vulnérabilités propres aux protocoles cryptographiques. Ces techniques visent à trouver des faiblesses dans des domaines tels que les longueurs de clé et les modes de chif-

frement qui peuvent présenter des risques de sécurité. Grâce à l'analyse des protocoles, l'analyse des protocoles de sécurité peut détecter des vulnérabilités susceptibles de conduire à un accès non autorisé, à des violations de l'intégrité des données ou à des atteintes à la vie privée. Cette analyse a le potentiel de fournir des orientations pour le développement et l'exécution de protocoles de communication sécurisés spécifiquement conçus pour répondre aux exigences particulières des systèmes automobiles.

Les méthodes formelles mentionnées ci-dessus, ainsi que d'autres techniques non divulguées, constituent un ensemble complet d'outils pour évaluer la sécurité des systèmes automobiles. Une évaluation complète de la posture de sécurité d'un système peut être réalisée en sélectionnant, en appliquant ou en combinant plusieurs approches formelles, en fonction des exigences spécifiques et des caractéristiques du système analysé. L'utilisation d'une stratégie globale qui capitalise sur les avantages de diverses approches formelles a le potentiel d'améliorer l'efficacité de l'analyse de sécurité et de faciliter la détection d'un large éventail de vulnérabilités. Cependant, il est impératif de reconnaître que l'utilisation de méthodes formelles nécessite un certain niveau de compétence et une évaluation minutieuse des complexités inhérentes au système. La nature complexe des systèmes automobiles et l'impératif d'inclure tous les facteurs de sécurité pertinents présentent des obstacles qui nécessitent une attention minutieuse lors de l'utilisation d'approches formelles pour l'examen de la sécurité des systèmes automobiles.

# 4 Techniques de validation pour assurer la sécurité des systèmes automobiles

Pour garantir la sécurité des systèmes automobiles, différentes techniques de validation sont utilisées. Cette section présente quelques techniques de validation couramment utilisées dans l'industrie, en consacrant une sous-section à chaque type.

## 4.1 Tests de pénétration

Les tests de pénétration, également appelés piratage éthique, consistent à simuler des attaques authentiques sur un système dans le but de détecter les vulnérabilités et d'évaluer l'efficacité des mesures de sécurité (13; 67; 104; 60; 231; 147). Les professionnels de la sécurité qualifiés s'efforcent d'exploiter les failles des mesures de défense du système, telles que les vulnérabilités logicielles ou les mauvaises configurations, dans le but d'obtenir un accès non autorisé ou d'exécuter des activités illégales. L'évaluation des vulnérabilités potentielles de sécurité dans les systèmes automobiles peut être réalisée grâce à la mise en œuvre de tests de pénétration, qui permettent d'identifier les faiblesses. Par la suite, des mesures appropriées peuvent être prises pour atténuer ces vulnérabilités.

### 4.1.1 Définitions mathématiques

Les tests de pénétration impliquent différentes méthodologies et techniques qui peuvent être définies mathématiquement et appliquées lors du processus de test. Celles-ci comprennent :

1. Évaluation des vulnérabilités (271; 114; 179) : Le processus d'identification et de quantification des vulnérabilités d'un système ou d'un réseau. Il consiste à analyser la configuration, le code et l'infrastructure du système pour découvrir d'éventuelles faiblesses.
2. Exploitation (171; 195; 251) : L'action de tirer parti d'une vulnérabilité ou d'une faiblesse du système pour obtenir un accès non autorisé ou effectuer des actions non autorisées. Cela peut impliquer l'exécution de code malveillant, la manipulation de données ou le contournement des contrôles de sécurité.
3. Élévation de privilèges (268) : Le processus d'élévation des privilèges utilisateur dans un système ou un réseau. Il consiste à exploiter les vulnérabilités pour obtenir des niveaux d'accès et de contrôle plus élevés sur le système ciblé.

### 4.1.2 Exemples de techniques de test de pénétration

Les tests de pénétration utilisent différentes techniques pour identifier les vulnérabilités et évaluer la posture de sécurité des systèmes automobiles. Voici quelques techniques courantes :

1. Analyse de réseau (276; 220) : Cette technique consiste à analyser le réseau cible pour identifier les hôtes actifs, les ports ouverts et les services s'exécutant sur ces ports. L'analyse de réseau permet d'identifier les points d'entrée potentiels et les systèmes vulnérables.
2. Crackage de mots de passe (108; 109) : Cette technique consiste à tenter de cracker des mots de passe pour obtenir un accès non autorisé à des comptes utilisateurs ou à des systèmes. Elle peut être réalisée à l'aide de différentes méthodes telles que les attaques par force brute, les attaques par dictionnaire ou les attaques par table arc-en-ciel.

Ce ne sont que quelques exemples des techniques utilisées dans les tests de pénétration. Le domaine des tests de pénétration est vaste et en constante évolution, avec l'émergence régulière de nouvelles techniques et outils pour découvrir et remédier aux vulnérabilités de sécurité dans les systèmes automobiles.

## 4.2 Injection de défauts

L'injection de défauts est une méthode utilisée pour évaluer la robustesse des systèmes automobiles en introduisant intentionnellement des défauts ou des erreurs (20; 62). En simulant divers scénarios de défauts, tels que des problèmes matériels, des erreurs logicielles ou des défaillances réseau, la capacité du système à gérer des circonstances imprévues ou atypiques peut être évaluée (74; 246). L'injection de défauts constitue une technique précieuse pour identifier les vulnérabilités, évaluer l'impact des défauts sur la sécurité du système et améliorer sa robustesse (103; 197).

### 4.2.1 Types d'injection de défauts

Il existe différents types de techniques d'injection de défauts pouvant être utilisées dans les tests de systèmes automobiles :

1. Injection de défauts matériels (77) : Cette technique consiste à introduire des défauts directement dans les composants matériels du système, tels que les microcontrôleurs, les capteurs ou les interfaces de communication. Par exemple, l'injection de surtensions ou d'interférences électromagnétiques peut simuler des conditions matérielles défectueuses et évaluer la résilience du système.
2. Injection de défauts logiciels (226) : En injectant des défauts dans les composants logiciels du système, tels que le système d'exploitation, les intergiciels ou les logiciels d'application, l'impact des erreurs logicielles sur le comportement du système peut être évalué. Des exemples incluent l'injection d'erreurs aléatoires dans le traitement des données ou la mise en évidence de vulnérabilités logicielles spécifiques pour tester la réponse du système.
3. Injection de défauts réseau (45; 45) : Cette technique consiste à injecter des défauts au niveau du réseau pour évaluer le comportement du système dans différentes conditions réseau. Par exemple, l'introduction de pertes de paquets, de latence ou de congestion réseau peut évaluer la capacité du système à gérer les défaillances de communication ou les conditions réseau défavorables.
4. Injection de défauts de synchronisation (238; 275) : Les défauts de synchronisation consistent à injecter des erreurs liées au temps et à la synchronisation dans le système. Cette technique vise à évaluer le comportement du système lorsqu'il est confronté à des violations de timing ou à des défaillances de synchronisation. Par exemple, l'injection de retards ou la modification du timing d'événements critiques peut évaluer la réponse du système et sa résilience aux défauts liés au timing.

## 4.3 Tests de Fuzzing

Les tests de fuzzing, également connus sous le nom de fuzz testing, sont une technique qui consiste à fournir à un système une grande quantité de données invalides, inattendues ou aléatoires afin de mettre en évidence des vulnérabilités ou des bugs logiciels (160; 155; 157; 119). En soumettant le système à de telles entrées, les tests de fuzzing visent à identifier les éventuelles faiblesses de sécurité, telles que les débordements de tampon ou les erreurs de validation des entrées (201; 190; 69; 70). Les tests de fuzzing peuvent être automatisés et améliorent considérablement la sécurité des systèmes automobiles en identifiant et en corrigeant les vulnérabilités logicielles (196; 85; 274; 259).

### 4.3.1 Fonctionnement des tests de Fuzzing

Les tests de fuzzing suivent un processus simple mais efficace :

1. Génération des entrées : Les tests de fuzzing consistent à générer une variété de données d'entrée qui peuvent potentiellement déclencher un comportement inattendu dans le système. Cela peut inclure des données malformées, des entrées

aléatoires ou des cas limites qui se situent en dehors de la plage normale des entrées valides.
2. Mutation des entrées : Les entrées générées sont ensuite mutées ou modifiées pour créer des variations supplémentaires. Cela permet d'explorer différents chemins et de mettre en évidence des vulnérabilités qui pourraient être sensibles à des motifs d'entrée spécifiques.
3. Injection des entrées : Les entrées mutées sont injectées dans le système à tester. Cela peut être réalisé en fournissant directement les entrées aux interfaces du système, telles que les API, les interfaces en ligne de commande ou les analyseurs de fichiers.
4. Surveillance et analyse : Pendant l'exécution du système avec les entrées fuzzées, des outils de surveillance et d'analyse sont utilisés pour détecter les anomalies, les plantages ou les comportements inattendus. Ces informations sont ensuite utilisées pour identifier les vulnérabilités ou les bugs potentiels.
5. Rapport de bugs et correction : Lorsqu'une vulnérabilité ou un bug est découvert grâce aux tests de fuzzing, il est signalé aux développeurs ou à l'équipe de sécurité responsable du système. Ils peuvent ensuite examiner le problème, le reproduire et appliquer les corrections appropriées pour améliorer la sécurité et la stabilité du système.

#### 4.3.2 Exemple de scénario de tests de Fuzzing

Prenons l'exemple où les tests de fuzzing sont appliqués à un module d'analyseur de fichiers dans un système automobile. L'objectif est d'identifier d'éventuelles vulnérabilités ou des plantages lors de l'analyse de fichiers d'entrée malformés.

Les tests de fuzzing permettent aux développeurs de systèmes automobiles et aux chercheurs en sécurité d'identifier et de corriger proactivement les vulnérabilités potentielles ou les bugs dans les composants logiciels. En soumettant systématiquement le système à des entrées inattendues, les tests de fuzzing contribuent à améliorer la sécurité et la fiabilité globales des systèmes automobiles.

### 4.4 Examen du code de sécurité

L'examen du code de sécurité consiste en l'inspection manuelle ou automatisée du code source afin d'identifier les faiblesses de sécurité, telles que les pratiques de codage non sécurisées ou les vulnérabilités (100; 93). En examinant attentivement la base de code, les experts en sécurité peuvent identifier des failles de sécurité potentielles, notamment une utilisation incorrecte des algorithmes cryptographiques, un manque de validation des entrées ou une protection inadéquate contre les vecteurs d'attaque courants (254; 202). Effectuer des examens approfondis du code de sécurité est essentiel pour garantir la robustesse des systèmes automobiles (198; 34).

#### 4.4.1 Avantages de l'examen du code de sécurité

L'examen du code de sécurité offre plusieurs avantages dans le développement et la maintenance des systèmes automobiles (53) :

- Détection des vulnérabilités : En examinant le code source, les experts en sécurité peuvent identifier des vulnérabilités et des faiblesses qui ne sont pas facilement détectables par d'autres techniques de test. Cela permet d'identifier et de réduire les risques de sécurité dès le début, avant qu'ils ne puissent être exploités.
- Identification des meilleures pratiques de sécurité : Les examens du code permettent de s'assurer que la base de code respecte les meilleures pratiques de sécurité de l'industrie. Cela inclut la vérification de l'utilisation correcte des algorithmes cryptographiques, de la validation sécurisée des entrées et de la protection contre les vulnérabilités de sécurité courantes, telles que les attaques par injection ou les vulnérabilités XSS (cross-site scripting).
- Conformité aux exigences réglementaires : Les systèmes automobiles sont souvent soumis à des exigences réglementaires et à des normes de l'industrie en matière de sécurité. Les examens du code de sécurité aident à garantir la conformité à ces exigences, réduisant ainsi le risque de sanctions et de conséquences juridiques.
- Partage des connaissances et collaboration en équipe : Les examens du code favorisent le partage des connaissances et la collaboration entre les équipes de développement. Ils offrent une opportunité aux experts en sécurité et aux développeurs d'échanger des idées, de traiter les problèmes de sécurité potentiels et d'améliorer leur compréhension des pratiques de codage sécurisé.
- Amélioration continue : Effectuer des examens du code de sécurité dans le cadre du processus de développement favorise une culture d'amélioration continue. En recherchant activement et en traitant les faiblesses de sécurité, l'équipe de développement peut renforcer la posture de sécurité du système automobile au fil du temps.

### 4.4.2 Approches de l'examen du code de sécurité

Les examens du code de sécurité peuvent être effectués selon différentes approches, notamment (15; 7) :

- Examen manuel du code : Dans un examen manuel du code, les experts en sécurité examinent attentivement le code source, ligne par ligne, pour identifier les faiblesses de sécurité. Cette approche nécessite une expertise en pratiques de codage sécurisées et une compréhension des vulnérabilités potentielles spécifiques aux systèmes automobiles.
- Analyse automatisée du code : Des outils et des scanners automatisés peuvent être utilisés pour effectuer une analyse statique du code et identifier les éventuelles failles de sécurité. Ces outils peuvent analyser rapidement la base de code, vérifier les vulnérabilités courantes et fournir une liste de problèmes potentiels. Cependant, ils peuvent également générer de faux positifs ou ignorer certaines vulnérabilités qui nécessitent un jugement humain.
- Combinaison d'approches manuelles et automatiques : Une combinaison d'approches d'examen manuel et automatisé du code est souvent utilisée pour maximiserl'efficacité de l'examen du code de sécurité. Les examens manuels permettent une compréhension approfondie du code source et une détection précise des faiblesses de sécurité spécifiques, tandis que les outils automatisés peuvent

accélérer le processus en identifiant les vulnérabilités courantes de manière plus rapide et systématique.

### 4.4.3 Meilleures pratiques pour les examens du code de sécurité

Pour garantir l'efficacité des examens du code de sécurité, il est recommandé de suivre certaines meilleures pratiques (50; 36) :

- Définir des critères d'examen clairs : Définissez des critères d'examen clairs avant de commencer l'examen du code. Cela peut inclure des directives sur les vulnérabilités spécifiques à rechercher, les bonnes pratiques de codage sécurisé et les exigences réglementaires à respecter.
- Utiliser des normes de codage sécurisé : Adoptez des normes de codage sécurisé, telles que les guides de sécurité OWASP (Open Web Application Security Project) ou les normes de sécurité applicables à l'industrie automobile. Ces normes fournissent des recommandations détaillées sur les bonnes pratiques de codage sécurisé et peuvent servir de référence lors de l'examen du code.
- Former les examinateurs en sécurité : Assurez-vous que les examinateurs du code de sécurité possèdent une expertise en matière de sécurité des systèmes automobiles et une connaissance approfondie des vulnérabilités courantes. Fournissez une formation adéquate sur les meilleures pratiques de codage sécurisé et les techniques d'examen du code.
- Documenter les résultats : Documentez soigneusement les résultats de l'examen du code de sécurité, y compris les vulnérabilités identifiées, les recommandations de correction et les actions prises pour remédier aux problèmes de sécurité. Cette documentation peut être utilisée pour suivre les progrès, évaluer la conformité et servir de référence pour les futurs examens du code.
- Intégrer l'examen du code dans le processus de développement : Intégrez l'examen du code de sécurité dans le processus de développement logiciel dès le début du projet. Cela garantit que les problèmes de sécurité sont traités dès le début et évite les retards ou les coûts supplémentaires liés à la correction de vulnérabilités tardivement dans le cycle de développement.
- Effectuer des examens réguliers : Réalisez des examens réguliers du code de sécurité tout au long du cycle de vie du système automobile. Les examens réguliers permettent de détecter et de traiter les nouvelles vulnérabilités introduites lors des mises à jour ou des modifications du code source.

### 4.4.4 Conclusion

L'examen du code de sécurité est une pratique essentielle pour garantir la robustesse et la sécurité des systèmes automobiles. En identifiant les faiblesses de sécurité potentielles dans le code source, les examens du code contribuent à réduire les risques de sécurité et à renforcer la posture de sécurité globale du système. En combinant des approches manuelles et automatisées, en suivant les meilleures pratiques et en intégrant l'examen du code dans le processus de développement, les équipes de développement peuvent s'assurer que les systèmes automobiles sont développés en respectant les normes de sécurité les plus élevées.

## 4.5 Modélisation des menaces

La modélisation des menaces est une approche systématique visant à identifier et à hiérarchiser les menaces potentielles pour le système, à évaluer leur impact et à concevoir des contre-mesures appropriées (261; 262). Cela implique d'analyser les composants du système, les interactions et les vecteurs d'attaque potentiels afin de comprendre les risques en matière de sécurité (267; 280). En identifiant et en hiérarchisant les menaces, les développeurs et les professionnels de la sécurité peuvent allouer efficacement les ressources nécessaires pour remédier aux vulnérabilités les plus critiques et améliorer la sécurité globale des systèmes automobiles (113; 105).

Dans le contexte des systèmes automobiles, la modélisation des menaces joue un rôle crucial pour garantir la sécurité des véhicules et de leurs technologies associées (232; 110). À mesure que la technologie continue de progresser, les véhicules deviennent de plus en plus interconnectés et dépendants des logiciels, ce qui les rend plus vulnérables aux menaces cybernétiques (89; 260). La modélisation des menaces aide les organisations à identifier et à atténuer de manière proactive les risques potentiels, réduisant ainsi la probabilité d'attaques réussies et leur impact potentiel.

Lors de la réalisation de la modélisation des menaces pour les systèmes automobiles, plusieurs considérations importantes entrent en jeu. Il est essentiel de comprendre l'architecture du système, y compris ses composants, ses interfaces et ses flux de données. Cette compréhension permet d'identifier les vecteurs d'attaque potentiels et les points faibles du système.

La modélisation des menaces implique généralement les étapes suivantes :

1. Identification des actifs : Commencez par identifier les actifs de valeur au sein du système automobile. Cela inclut non seulement le véhicule lui-même, mais également les données qu'il génère et traite, telles que les informations personnelles, les données de navigation et la télémétrie du véhicule.
2. Création d'une vue d'ensemble du système : Développez une compréhension complète de l'architecture du système, y compris les composants matériels, logiciels et réseau. Cette étape consiste à cartographier les différents éléments du système, leurs relations et le flux d'information entre eux.
3. Identification des menaces : Une fois l'architecture du système comprise, identifiez systématiquement les menaces potentielles et les vulnérabilités. Cela peut être réalisé en envisageant des scénarios d'attaque potentiels, en analysant des modèles d'attaques historiques et en s'appuyant sur les meilleures pratiques de l'industrie et les directives de sécurité.
4. Évaluation de l'impact : Évaluez l'impact potentiel de chaque menace identifiée sur le système et ses actifs. Prenez en compte des facteurs tels que la probabilité que la menace soit exploitée, les conséquences potentielles d'une attaque réussie et les risques associés à la sécurité, à la confidentialité et aux aspects financiers.
5. Hiérarchisation des contre-mesures : Hiérarchisez les menaces identifiées en fonction de leur impact potentiel et de leur probabilité d'occurrence. Cette étape permet d'allouer efficacement les ressources, en veillant à ce que les vulnérabilités les plus critiques soient traitées en premier. Il est important d'impliquer les parties prenantes concernées, notamment les développeurs, les ingénieurs et les

professionnels de la sécurité, dans ce processus afin d'obtenir des perspectives et une expertise diverses.

6. Développement des contre-mesures : Une fois les menaces hiérarchisées, concevez les contre-mesures appropriées pour atténuer les risques identifiés. Cela peut inclure la mise en place de contrôles de sécurité, l'application de bonnespratiques de codage sécurisé, la réalisation de tests de pénétration et l'établissement de plans d'intervention en cas d'incident. Il est crucial de prendre en compte à la fois les contre-mesures techniques et les contre-mesures procédurales pour garantir une approche holistique de la sécurité.
7. Surveillance et amélioration continues : La modélisation des menaces n'est pas un processus ponctuel, mais plutôt un effort itératif et continu. À mesure que de nouvelles menaces émergent et que le système évolue, il est important de réévaluer et de mettre à jour régulièrement le modèle de menace. Cela garantit que le système reste résistant aux menaces de sécurité en constante évolution tout au long de son cycle de vie.

En incorporant la modélisation des menaces dans le développement et la maintenance des systèmes automobiles, les organisations peuvent aborder de manière proactive les vulnérabilités de sécurité et améliorer la résilience globale de leurs produits. Cette approche permet de minimiser le potentiel d'attaques réussies, de protéger la sécurité et la confidentialité des occupants des véhicules, ainsi que de prévenir l'accès non autorisé et l'utilisation abusive de données et d'infrastructures critiques.

## 4.6 Cadres de tests de sécurité

L'utilisation de cadres spécialisés fournissant des outils et des méthodologies pour mener des tests de sécurité complets est essentielle (18; 205; 175). Ces cadres englobent un éventail de techniques, y compris la recherche de vulnérabilités, l'analyse de code et l'évaluation de la sécurité (102; 175; 206). Ils aident à automatiser les processus de tests de sécurité, à identifier les faiblesses en matière de sécurité et à garantir le respect des normes de sécurité établies (203; 173; 243). Les cadres de tests de sécurité fournissent une approche structurée et systématique pour évaluer la posture de sécurité des systèmes automobiles (165; 118; 174; 64).

### 4.6.1 Importance des cadres de tests de sécurité

Les cadres de tests de sécurité jouent un rôle essentiel dans l'évaluation de la sécurité des systèmes automobiles. Voici quelques raisons clés pour lesquelles les cadres de tests de sécurité sont importants :

— Tests complets : Les cadres de tests de sécurité fournissent un ensemble complet d'outils et de techniques pour évaluer la sécurité des systèmes automobiles. Ces cadres couvrent un large éventail d'aspects de sécurité, notamment la recherche de vulnérabilités, les tests de pénétration, l'analyse de code et l'évaluation de la sécurité. En utilisant ces cadres, les organisations peuvent effectuer des tests de sécurité approfondis et identifier les vulnérabilités potentielles et les faiblesses du système.

- Automatisation et efficacité : Les cadres de tests de sécurité automatisent divers processus de tests de sécurité, permettant aux organisations d'effectuer des tests de manière plus efficace et plus efficiente. Ces cadres fournissent des outils automatisés pour la recherche de vulnérabilités, l'analyse de code et d'autres activités de tests de sécurité. L'automatisation permet de réduire les efforts manuels, d'accélérer le processus de test et d'améliorer l'exactitude des évaluations de sécurité.
- Conformité aux normes : Les cadres de tests de sécurité intègrent souvent des normes de sécurité établies et des meilleures pratiques. Ils fournissent des lignes directrices et des vérifications pour garantir le respect de ces normes, telles que ISO 27001, NIST Cybersecurity Framework ou des exigences de sécurité spécifiques à l'industrie. En utilisant ces cadres, les organisations peuvent évaluer leur conformité aux normes de sécurité pertinentes et démontrer leur engagement envers la sécurité.
- Atténuation des risques : Les cadres de tests de sécurité aident à identifier les vulnérabilités et les faiblesses des systèmes automobiles, permettant aux organisations de les traiter de manière proactive. En effectuant régulièrement des tests de sécurité à l'aide de ces cadres, les organisations peuvent identifier et atténuer les risques potentiels avant qu'ils ne soient exploités par des attaquants. Cela contribue à réduire la probabilité et l'impact des violations de sécurité, protégeant ainsi les données sensibles et maintenant l'intégrité globale du système.
- Amélioration continue : Les cadres de tests de sécurité favorisent une approche d'amélioration continue de la sécurité. Ils fournissent aux organisations une méthode structurée et systématique pour évaluer la posture de sécurité de leurs systèmes automobiles de manière continue. En utilisant régulièrement ces cadres, les organisations peuvent identifier les menaces émergentes, traiter les vulnérabilités en évolution et améliorer globalement la sécurité de leurs systèmes au fil du temps.

### 4.6.2 Cadres courants de tests de sécuritéI

l existe plusieurs cadres de tests de sécurité largement utilisés pour évaluer la sécurité des systèmes automobiles. Voici quelques exemples :

- OWASP Testing Guide (178; 83) : The Open Web Application Security Project (OWASP) fournit un guide de tests complet qui couvre différents aspects de la sécurité des applications. Il comprend des méthodologies, des outils et des techniques pour tester les applications web, les API et d'autres composants logiciels.
- NIST SP 800-115 (2) : La publication spéciale 800-115 de l'Institut national des normes et de la technologie (NIST) fournit des conseils sur les tests et l'évaluation de la sécurité de l'information. Elle couvre des sujets tels que les tests de pénétration, la recherche de vulnérabilités et les méthodologies d'évaluation de la sécurité.
- OSSTMM (68) : Le Manuel de méthodologie de test de sécurité open source (OSSTMM) est un cadre qui fournit des lignes directrices et des méthodologies

pour les tests de sécurité. Il couvre des domaines tels que la sécurité réseau, la sécurité physique et la sécurité opérationnelle.
— PTES (3) : Le Penetration Testing Execution Standard (PTES) est un cadre qui fournit une approche normalisée pour la réalisation de tests de pénétration. Il comprend une méthodologie et des lignes directrices pour effectuer des tests de pénétration approfondis sur différents systèmes et applications.
— ISSAF (1) : Le cadre d'évaluation de la sécurité des systèmes d'information (ISSAF) est un cadre qui fournit des conseils sur l'évaluation et les tests de sécurité. Il couvre des domaines tels que l'évaluation des risques, l'évaluation des vulnérabilités et l'audit de sécurité.

Ces cadres offrent une gamme d'outils, de méthodologies et de lignes directrices que les organisations peuvent exploiter pour évaluer efficacement la sécurité de leurs systèmes automobiles.

### 4.6.3 Intégration avec le cycle de développement

Pour maximiser l'efficacité des cadres de tests de sécurité, il est essentiel d'intégrer les tests de sécurité tout au long du cycle de développement des systèmes automobiles. En intégrant les tests de sécurité à chaque phase, de la collecte des exigences au déploiement, les organisations peuvent identifier et résoudre les problèmes de sécurité dès le début. L'intégration des cadres de tests de sécurité aux méthodologies de développement telles que DevSecOps permet de garantir que la sécurité est prise en compte à chaque étape du cycle de vie du système.

En utilisant des cadres de tests de sécurité, les organisations peuvent améliorer la sécurité de leurs systèmes automobiles, identifier les vulnérabilités et établir une posture de sécurité robuste qui protège contre les menaces potentielles.

## 4.7 Techniques de la Blockchain

Une blockchain est une chaîne croissante de blocs de données liés les uns aux autres. Les registres distribués sont répartis sur des réseaux peer-to-peer comme ce réseau de blocs de données. Les données numériques sont synchronisées, copiées, distribuées et partagées à travers un réseau peer-to-peer dans un registre distribué. Chaque partenaire du réseau possède la même copie du registre partagé, car chaque appareil possède la version la plus récente. La base de données ne peut être étendue qu'en ajoutant des blocs à la chaîne et le registre est sécurisé. Les modifications apportées aux enregistrements enregistrés dans la chaîne sont computationnellement impossibles. La décentralisation est l'un des principaux avantages du registre distribué. Aucune autorité centrale ne contrôle le registre, mais chaque nœud met à jour son registre lorsqu'un nouveau bloc est ajouté à la blockchain via une procédure de consensus commune. Les techniques de la blockchain ont suscité une attention considérable ces dernières années en raison de leur potentiel pour renforcer la sécurité et la confiance dans divers domaines (193; 51; 135; 161; 58; 151). Dans le contexte de la sécurité des systèmes automobiles, la blockchain peut être utilisée pour fournir des enregistrements de transactions inviolables et transparents, une gestion décentralisée des identités et des canaux de communication sécurisés (96; 234; 99; 218; 97; 71; 98).

### 4.7.1 Intégrité et traçabilité des mises à jour de logiciels

Une application de la blockchain dans la sécurité des systèmes automobiles consiste à garantir l'intégrité et la traçabilité des mises à jour de logiciels. En utilisant un registre distribué, les fabricants automobiles peuvent enregistrer et vérifier en toute sécurité l'authenticité des mises à jour de logiciels, empêchant ainsi les modifications non autorisées ou la falsification. Cela renforce la sécurité du système automobile en veillant à ce que seules des mises à jour de logiciels autorisées et vérifiées soient appliquées, réduisant ainsi le risque d'injection de code malveillant ou de modifications non autorisées.

### 4.7.2 Accords sécurisés avec des contrats intelligents

Des contrats intelligents basés sur la blockchain peuvent être utilisés pour établir des accords sécurisés et automatisés entre différentes entités de l'écosystème automobile, telles que les fournisseurs, les fabricants et les prestataires de services (189; 138; 95). Les contrats intelligents sont des contrats auto-exécutables dont les termes sont directement écrits dans le code. En tirant parti de la nature décentralisée et inviolable de la blockchain, les contrats intelligents peuvent faciliter des transactions sécurisées et transparentes, automatiser les processus de paiement et garantir le respect des règles et conditions prédéfinies (217; 131; 247). Cela contribue à rationaliser les interactions, à réduire la dépendance aux intermédiaires et à renforcer la sécurité et l'efficacité globales des opérations commerciales dans l'industrie automobile (101; 159; 41).

### 4.7.3 Partage sécurisé et préservation de la confidentialité des données

La technologie de la blockchain peut également être utilisée pour le partage sécurisé et la préservation de la confidentialité des données entre les véhicules et les composants d'infrastructure. Des techniques telles que les preuves de connaissance nulle et les transactions privées peuvent être utilisées pour garantir que des données sensibles, telles que des informations de localisation ou des diagnostics de véhicules, peuvent être partagées de manière sécurisée sans compromettre la confidentialité. LesLa technologie de la blockchain peut également être utilisée pour le partage sécurisé et la préservation de la confidentialité des données entre les véhicules et les composants d'infrastructure. Des techniques telles que les preuves de connaissance nulle et les transactions privées peuvent être utilisées pour garantir que des données sensibles, telles que des informations de localisation ou des diagnostics de véhicules, peuvent être partagées de manière sécurisée sans compromettre la confidentialité. Les preuves de connaissance nulle permettent la vérification d'une déclaration sans révéler les données sous-jacentes, tandis que les transactions privées garantissent que les détails des transactions ne sont visibles que par les parties autorisées. En tirant parti de la blockchain pour le partage sécurisé des données, les systèmes automobiles peuvent bénéficier d'une collaboration améliorée, d'une conscience situationnelle accrue et de services basés sur les données plus efficaces, tout en préservant la confidentialité et la sécurité des données.

## 4.8 Techniques d'apprentissage automatique

L'ingénierie logicielle consiste à écrire manuellement des instructions informatiques. L'automatisation de la rédaction de règles est ajoutée par l'apprentissage automatique. En d'autres termes, les développeurs de logiciels utilisent leur intelligence pour résoudre un problème et créer un programme informatique. Les scientifiques des données qui installent des systèmes d'apprentissage automatique n'écrivent pas leurs propres programmes. Ils collectent des données d'entrée et des valeurs cibles. Ils dirigent un ordinateur pour trouver un logiciel qui calcule les sorties pour chaque valeur d'entrée.

Les techniques d'apprentissage automatique sont de plus en plus utilisées dans la sécurité des systèmes automobiles pour détecter et atténuer les menaces de sécurité en temps réel (183; 8; 126; 9). Les algorithmes d'apprentissage automatique peuvent analyser de vastes quantités de données collectées à partir de capteurs divers, du trafic réseau et des journaux système pour identifier des comportements anormaux et des violations potentielles de sécurité (17; 209; 38; 127).

Dans le contexte de la cybersécurité automobile, l'apprentissage automatique peut être utilisé pour les systèmes de détection et de prévention des intrusions (200; 187; 16). En entraînant des modèles sur une combinaison d'activités normales et malveillantes, les algorithmes d'apprentissage automatique peuvent apprendre des modèles et identifier des écarts qui peuvent indiquer une attaque en cours (29; 26; 86; 213; 142; 264). Ces algorithmes peuvent surveiller en continu le réseau, détecter des activités suspectes et déclencher des mesures de sécurité appropriées (172; 40; 257).

De plus, les techniques d'apprentissage automatique peuvent être utilisées pour la détection d'anomalies dans le comportement des véhicules (65; 258). En établissant les lignes de base du fonctionnement normal du véhicule, les modèles d'apprentissage automatique peuvent identifier les écarts qui peuvent indiquer un accès non autorisé ou des tentatives de contrôle malveillantes (82; 224). Cela peut aider à protéger les véhicules contre les attaques cyber-physiques, telles que le piratage à distance ou la manipulation des systèmes critiques (235; 241).

De plus, les techniques d'apprentissage automatique peuvent jouer un rôle crucial dans la sécurisation des véhicules connectés et autonomes en permettant la maintenance prédictive et l'évaluation des vulnérabilités. En analysant les données des capteurs, les algorithmes d'apprentissage automatique peuvent identifier des schémas et des corrélations indiquant des vulnérabilités potentielles ou des défaillances imminentes dans les composants du véhicule. Cette approche proactive permet aux fabricants et aux prestataires de services de résoudre ces problèmes avant qu'ils ne puissent être exploités par des acteurs malveillants (5; 253).

De plus, l'apprentissage automatique peut être utilisé pour les mises à jour sécurisées du micrologiciel et des logiciels dans les véhicules. En utilisant des algorithmes de détection d'anomalies, les modèles d'apprentissage automatique peuvent vérifier l'intégrité et l'authenticité des mises à jour logicielles, en veillant à ce que seules les mises à jour autorisées et exemptes de manipulation soient installées. Cela élimine le risque d'introduction de logiciels compromis ou malveillants dans les systèmes du véhicule, garantissant ainsi la sécurité et la fiabilité dela pile logicielle du véhicule (87).

L'apprentissage automatique peut également améliorer l'efficacité des systèmes de réponse aux intrusions dans la cybersécurité automobile. En analysant et en apprenant

en continu des incidents de sécurité, les modèles d'apprentissage automatique peuvent améliorer leur capacité à détecter et à répondre aux menaces émergentes. Ces modèles peuvent s'adapter et évoluer avec le temps, en incorporant de nouvelles connaissances et techniques pour améliorer la posture globale de sécurité du véhicule (39).

En fin de compte, les techniques d'apprentissage automatique offrent un potentiel significatif pour améliorer la sécurité des systèmes automobiles. Elles permettent la détection des menaces en temps réel, la détection d'anomalies, la maintenance prédictive, les mises à jour sécurisées et l'amélioration de la réponse aux intrusions. Alors que l'industrie automobile continue d'adopter les technologies de connectivité et d'autonomie, l'intégration de l'apprentissage automatique dans les pratiques de cybersécurité deviendra de plus en plus cruciale pour garantir la sécurité des véhicules et de leurs occupants (5; 253).

# 5 Intégration des méthodes formelles et des techniques de validation

Assurer la sécurité des systèmes automobiles nécessite une approche globale qui intègre les méthodes formelles et les techniques de validation tout au long du cycle de développement. Cette section examine comment ces méthodologies peuvent être intégrées de manière efficace dans les phases clés du développement des systèmes automobiles.

## 5.1 Phase d'ingénierie des exigences

Dans la phase d'ingénierie des exigences, les bases de la sécurité du système sont établies de manière méticuleuse. Cette phase constitue un point critique où les objectifs de sécurité fondamentaux et les contraintes sont définis. Les méthodes formelles apparaissent comme des outils indispensables dans ce processus, apportant une approche structurée et rigoureuse pour spécifier les exigences de sécurité.

Les langages de spécification formelle, tels que la notation Z et Alloy, ainsi que les notations basées sur les modèles comme UMLsec, se situent au premier plan de cette entreprise. Ces outils fournissent un cadre systématique pour exprimer les propriétés de sécurité, garantissant qu'elles sont articulées avec précision et clarté. En utilisant les méthodes formelles, les ambiguïtés et les interprétations erronées potentielles qui peuvent survenir dans les spécifications en langage naturel sont atténuées. Cela favorise une compréhension partagée entre les parties prenantes, des développeurs aux experts en sécurité, jetant ainsi les bases solides des phases ultérieures.

De plus, les méthodes formelles aident à l'identification et à la représentation des propriétés de sécurité critiques. Celles-ci peuvent englober la confidentialité, l'intégrité, la disponibilité et d'autres attributs essentiels. Grâce à l'application de techniques formelles, les subtilités des exigences de sécurité sont disséquées, permettant une définition complète et non ambiguë de la posture de sécurité du système.

Les techniques de validation complètent les méthodes formelles pendant cette phase, enrichissant le processus d'une perspective pratique et axée sur les risques. Des activités telles que la modélisation des menaces et l'analyse des risques prennent le devant de la

scène. La modélisation des menaces évalue systématiquement les menaces potentielles et les vulnérabilités auxquelles le système peut être confronté. En examinant les vecteurs d'attaque et les scénarios d'exploitation potentiels, l'équipe de développement acquiert des informations inestimables sur le paysage de sécurité du système automobile.

Les conclusions de la modélisation des menaces et de l'analyse des risques se combinent avec le processus de spécification formelle. Elles fournissent une source d'informations contextuelles, permettant une définition plus précise des exigences de sécurité. Cette relation symbiotique entre les techniques de validation et les méthodes formelles aboutit à un ensemble de spécifications de sécurité affinées et complètes.

En fin de compte, la phase d'ingénierie des exigences incarne une intégration harmonieuse des méthodes formelles et des techniques de validation. Les méthodes formelles fournissent la base de spécifications de sécurité structurées et non ambiguës, tandis que les techniques de validation apportent une compréhension pratique des risques de sécurité du monde réel. Ensemble, ils posent la première pierre d'une approche résiliente et bien informée de la sécurité des systèmes automobiles.

## 5.2 Phase de conception

La phase de conception représente une étape cruciale dans le développement des systèmes automobiles, où la conception de l'architecture du système est élaborée. Les méthodes formelles continuent de jouer un rôle central dans cette phase, offrant une approche structurée pour affiner l'architecture de sécurité.

La vérification de modèles, un pilier de la vérification formelle, revêt une importance particulière pendant la phase de conception. Cette technique permet une exploration exhaustive du comportement du système par rapport aux propriétés de sécurité spécifiées. En soumettant la conception à une batterie de scénarios méticuleusement élaborés, la vérification de modèles révèle des défauts ou des incohérences potentiels dans la conception. Il peut s'agir de problèmes liés au contrôle d'accès, à la circulation des données ou à d'autres aspects critiques de la sécurité. Identifier ces vulnérabilités à ce stade précoce est primordial, car cela permet de mettre en œuvre des mesures correctives préventives avant que le système ne progresse davantage.

En parallèle avec les méthodes formelles, les techniques de validation interviennent pour fournir une évaluation pratique et pratique de la robustesse de la sécurité de la conception. L'examen de l'architecture de sécurité, une examination minutieuse des composants architecturaux du système, joue un rôle central dans ce processus d'évaluation. Cette activité examine en détail des éléments clés tels que les configurations réseau, les protocoles cryptographiques et les contrôles d'accès. En disséquant l'architecture d'un point de vue de la sécurité, des faiblesses ou des omissions potentielles sont mises au jour, apportant des informations précieuses pour l'affinage.

De plus, la modélisation des menaces continue d'être un outil essentiel pendant la phase de conception. Ce processus structuré implique l'identification et l'évaluation des menaces et vulnérabilités potentielles dans la conception du système. En simulant des scénarios d'attaque et en envisageant des agents de menace potentiels, les développeurs acquièrent une compréhension globale du paysage de sécurité. Les informations tirées de la modélisation des menaces sont essentielles pour affiner l'architecture de sécurité, garantissant qu'elle reste résiliente face à un éventail de menaces potentielles.

L'interaction entre les méthodes formelles et les techniques de validation pendant la phase de conception aboutit à une architecture de sécurité solide et méticuleusement élaborée. Les méthodes formelles mettent en évidence les subtilités de la conception, tandis que les techniques de validation offrent une vérification pratique par rapport aux menaces potentielles du monde réel. Ensemble, elles créent une conception qui répond non seulement aux exigences fonctionnelles, mais qui témoigne également d'un engagement ferme envers la sécurité.

## 5.3 Phase de mise en œuvre

La phase de mise en œuvre marque une transition cruciale dans le processus de développement, où les conceptions et spécifications abstraites sont concrètement réalisées sous forme de code exécutable. Les méthodes formelles prennent le devant de la scène pendant cette phase, veillant à ce que les exigences de sécurité soigneusement définies soient fidèlement traduites en code respectant les propriétés de sécurité spécifiées.

La démonstration de théorèmes constitue une technique redoutable dans cette phase, offrant un processus de vérification mathématique rigoureux. Elle soumet le code à une série de preuves formelles, examinant méticuleusement chaque ligne pour vérifier sa conformité aux propriétés de sécurité spécifiées. Ce processus offre un haut degré de confiance que le code implémenté est conforme à la posture de sécurité souhaitée, réduisant ainsi le risque de vulnérabilités accidentelles.

En parallèle, les techniques de validation jouent un rôle clé dans la phase de mise en œuvre. Parmi celles-ci, l'examen du code de sécurité revêt une importance primordiale. Ce processus d'examen systématique implique une inspection approfondie du code à la recherche de potentielles vulnérabilités de sécurité. Les développeurs inspectent méticuleusement le code, identifiant les points faibles potentiels ou les omissions qui pourraient exposer lesystème à des risques de sécurité. En corrigeant ces problèmes à ce stade, les développeurs renforcent préventivement le système contre les menaces potentielles, réduisant ainsi la probabilité que des vulnérabilités se manifestent dans l'environnement déployé.

De plus, les cadres de test de sécurité, intégrés dans la phase de mise en œuvre, jouent un rôle vital dans la validation de la robustesse de sécurité du code mis en œuvre. Ces cadres soumettent le code à une série de scénarios d'attaque simulés, évaluant sa résilience face aux menaces potentielles.

La synergie entre les méthodes formelles et les techniques de validation pendant la phase de mise en œuvre représente une solide protection contre les vulnérabilités de sécurité. Les méthodes formelles instaurent une rigueur mathématique, garantissant que le code est conforme aux propriétés de sécurité spécifiées. Simultanément, les techniques de validation, en mettant l'accent sur l'examen du code de sécurité, offrent une protection pratique contre les vulnérabilités potentielles. Ensemble, elles créent une mise en œuvre qui non seulement satisfait aux exigences fonctionnelles, mais qui témoigne également d'un engagement solide envers la sécurité.

## 5.4 Phase de test

La phase de test constitue un moment critique où l'efficacité des mesures de sécurité est rigoureusement évaluée. Cette phase représente la culmination de l'intégration des méthodes formelles et des techniques de validation, visant à affirmer la robustesse de la posture de sécurité du système automobile.

Les méthodes formelles continuent d'exercer leur puissance analytique pendant cette phase. Des techniques telles que la vérification de modèles et l'interprétation abstraite sont utilisées pour examiner systématiquement le comportement du système par rapport aux propriétés de sécurité spécifiées. En particulier, la vérification de modèles permet une exploration exhaustive des états et des transitions potentiels, offrant l'assurance que les propriétés de sécurité critiques sont respectées dans différentes conditions. L'interprétation abstraite complète cela en fournissant une analyse plus large du comportement du système, permettant l'identification de vulnérabilités potentielles qui pourraient avoir échappé à la vérification formelle seule.

Parallèlement, les techniques de validation émergent en tant qu'agents pratiques de résilience de sécurité. Les tests de pénétration, pierre angulaire de l'évaluation de sécurité, consistent en des attaques simulées sur le système pour mettre en évidence d'éventuelles vulnérabilités. En reproduisant des scénarios d'attaque réels, les tests de pénétration offrent une vérification cruciale de la réalité, révélant des faiblesses qui n'auraient pas été évidentes par le biais des processus de vérification formelle seuls. L'injection de défauts, une autre technique puissante, consiste à introduire délibérément des erreurs ou des fautes dans le système pour évaluer sa résilience. Cette méthode simule des circonstances imprévues, offrant des informations sur le comportement du système dans des conditions défavorables. De plus, les tests de fuzzing introduisent des entrées inattendues dans le système, sondant les vulnérabilités qui pourraient découler de données imprévues.

L'intégration de ces techniques de validation dans la phase de test améliore considérablement l'exhaustivité de l'évaluation de sécurité. Elles introduisent une dimension pratique, soumettant le système à des scénarios d'attaque réels, mettant ainsi en évidence des vulnérabilités potentielles que les méthodes formelles seules pourraient ne pas détecter.

Grâce à l'intégration transparente des méthodes formelles et des techniques de validation, la phase de test constitue l'épreuve ultime de la sécurité du système automobile. Les méthodes formelles offrent une approche structurée et analytique de la vérification de sécurité, tandis que les techniques de validation valident de manière pratique les mesures de sécurité dans des conditions réelles. Ensemble, elles garantissent que le système automobile sort de cette phase avec une posture de sécurité solide et validée.

En intégrant les méthodes formelles et les techniques de validation à ces phases clés, les systèmes automobiles peuvent atteindre un niveau de robustesse de sécurité plus élevé. Cette approche holistique garantit que les considérations de sécurité sont intégrées dans le processus de développement, donnant ainsi naissance à des systèmes plus résilients et mieux préparés à faire face aux menaces cybernétiques en constante évolution.

## 5.5 Aspects distinctifs des méthodes formelles et des techniques de validation pour renforcer la sécurité des systèmes automobiles

Les méthodes formelles et les techniques de validation jouent un rôle crucial dans l'amélioration de la sécurité de différents systèmes, y compris les systèmes automobiles, les appareils IoT, les applications logicielles et les systèmes embarqués. Bien qu'il existe des similitudes dans l'application de ces méthodes dans différents domaines, il existe également des aspects distinctifs spécifiques à l'utilisation des méthodes formelles et des techniques de validation pour la sécurité des systèmes automobiles :

1. Complexité et caractère critique pour la sécurité des systèmes automobiles : Les systèmes automobiles se caractérisent par leur complexité et leur caractère critique pour la sécurité. Ils impliquent des interactions complexes entre différents composants, tels que les capteurs, les actionneurs, les unités de contrôle et les réseaux de communication. Les méthodesformelles et les techniques de validation doivent relever les défis uniques posés par la complexité des systèmes automobiles, tels que la modélisation du comportement des composants interconnectés, la vérification des propriétés de sécurité et l'assurance de la fiabilité et de la robustesse du système dans différentes conditions de fonctionnement.
2. Contraintes temporelles réelles et exigences de performance : Les systèmes automobiles fonctionnent dans des environnements en temps réel, où des réponses rapides et précises sont essentielles pour garantir la sécurité. Les méthodes formelles et les techniques de validation pour les systèmes automobiles doivent prendre en compte les contraintes temporelles réelles, telles que les temps de réponse, la latence et les exigences de synchronisation. Analyser et vérifier le comportement temporel des systèmes automobiles est crucial pour prévenir d'éventuelles vulnérabilités de sécurité et assurer le bon fonctionnement du système.
3. Intégration des considérations de sécurité et de sûreté : Contrairement à d'autres systèmes, les systèmes automobiles nécessitent l'intégration à la fois des considérations de sécurité et de sûreté. Alors que la sûreté vise à prévenir les accidents et à minimiser les dommages aux occupants et aux piétons, la sécurité concerne la protection du système contre les attaques malveillantes et les accès non autorisés. Les méthodes formelles et les techniques de validation dans le domaine automobile doivent englober à la fois les aspects de sûreté et de sécurité, en veillant à ce que le système soit résistant à la fois aux défaillances accidentelles et aux attaques intentionnelles.
4. Paysage des menaces spécifique à l'automobile : Le domaine automobile présente un paysage de menaces unique par rapport à d'autres systèmes. Les systèmes automobiles sont vulnérables à un large éventail de menaces de sécurité, telles que les exploitations à distance, l'accès non autorisé au réseau du véhicule, la manipulation des unités de contrôle électronique (ECU) et la compromission de l'intégrité des données des capteurs. Les méthodes formelles et les techniques de validation pour la sécurité des systèmes automobiles doivent prendre en compte ces menaces et vulnérabilités spécifiques, en tenant compte de l'impact potentiel sur la sécurité, la confidentialité et la fonctionnalité globale du véhicule.

5. Conformité aux normes et réglementations de l'industrie : L'industrie automobile est soumise à des normes et réglementations strictes en matière de sécurité et de sûreté. Les méthodes formelles et les techniques de validation doivent être conformes à ces normes spécifiques à l'industrie, telles que ISO 26262 pour la sécurité fonctionnelle et ISO/SAE 21434 pour la cybersécurité automobile. Le respect de ces normes garantit que l'application des méthodes formelles et des techniques de validation dans les systèmes automobiles répond aux exigences et aux lignes directrices nécessaires en matière de sécurité et de sûreté.

En résumé, bien qu'il existe des similitudes dans l'application des méthodes formelles et des techniques de validation dans différents domaines, l'utilisation de ces méthodes pour renforcer la sécurité des systèmes automobiles comporte des aspects distinctifs. La complexité et le caractère critique pour la sécurité des systèmes automobiles, les contraintes temporelles réelles, l'intégration des considérations de sécurité et de sûreté, le paysage des menaces spécifique à l'automobile et la conformité aux normes de l'industrie contribuent tous aux défis et aux considérations uniques liés à la sécurisation des systèmes automobiles. Aborder ces aspects distinctifs est crucial pour appliquer efficacement les méthodes formelles et les techniques de validation afin d'améliorer la sécurité des systèmes automobiles.

# 6 Avantages et limitations des approches

L'intégration des méthodes formelles et des techniques de validation apporte de nombreux avantages dans le domaine de la sécurité des systèmes automobiles. Cependant, il est important de reconnaître que ces approches ne sont pas sans considérations et contraintes propres.

## 6.1 Considérations de scalabilité

Avantages : La scalabilité est un facteur crucial dans l'évaluation des méthodologies de sécurité, surtout dans le contexte des systèmes automobiles complexes. Les méthodes formelles, avec leurs fondements mathématiques, excellent dans la gestion de la complexité. Elles offrent la possibilité d'analyser des systèmes complexes avec précision, en veillant à ce que les propriétés de sécurité soient vérifiées de manière exhaustive. Cela s'avère inestimable pour identifier les vulnérabilités potentielles dans les systèmes automobiles à grande échelle.

Limitations : Cependant, à mesure que les systèmes deviennent de plus en plus complexes, les ressources informatiques requises pour la vérification formelle peuvent augmenter considérablement. Par exemple, la vérification de modèles peut rencontrer des défis de scalabilité lorsqu'elle traite des espaces d'états exceptionnellement grands. Cela rend nécessaire une réflexion minutieuse sur l'allocation des ressources et l'exploration de techniques spécialisées pour aborder les problèmes de scalabilité.

### 6.2 Considérations d'efficacité

Avantages : Les méthodes formelles offrent un niveau d'assurance difficile à égaler avec des tests purement empiriques. Elles offrent une approche systématique et exhaustive de la vérification de sécurité. En exploitant la rigueur mathématique, les méthodes formelles peuvent identifier les vulnérabilités avec un degré de confiance élevé, réduisant ainsi la probabilité de faux négatifs.

Limitations : Cependant, les méthodes formelles peuvent être intensives en termes de calcul, ce qui peut entraîner des temps de vérification prolongés. Par exemple, la démonstration de théorèmes peut nécessiter des ressources informatiques importantes pour établir la correction de segments de code complexes. Trouver un équilibre entre précision et efficacité est primordial pour s'assurer que le processus de vérification reste praticable dans les contraintes des cycles de développement du monde réel.

### 6.3 Applicabilité aux systèmes automobiles du monde réel

Avantages : L'efficacité des méthodes formelles et des techniques de validation dans le contexte des systèmes automobiles du monde réel est évidente dans leur capacité à découvrir des vulnérabilités de sécurité subtiles. En soumettant les systèmes à une analyse rigoureuse et à des scénarios d'attaques réelles, ces approches offrent une défense solide contre les menaces potentielles. Cette approche proactive de la sécurité est conforme à l'évolution du paysage technologique automobile, où les menaces cybernétiques continuent d'évoluer.

Limitations : Cependant, l'applicabilité des méthodes formelles et des techniques de validation peut varier en fonction des caractéristiques spécifiques du système automobile. Des systèmes hautement spécialisés ou propriétaires peuvent présenter des défis uniques en termes d'intégration et d'adéquation à certaines techniques de vérification formelle. De plus, la disponibilité de praticiens qualifiés maîtrisant les méthodes formelles peut influencer la faisabilité de leur adoption généralisée dans l'industrie automobile.

En résumé, l'intégration des méthodes formelles et des techniques de validation offre un arsenal puissant dans la recherche de la sécurité des systèmes automobiles. Bien qu'elles offrent une précision inégalée dans l'identification des vulnérabilités, il est nécessaire de prendre en compte les considérations de scalabilité, d'efficacité et d'applicabilité aux systèmes du monde réel. Trouver un équilibre entre ces facteurs est crucial pour exploiter pleinement le potentiel de ces méthodologies.

## 7 Tendances actuelles de la recherche et questions de recherche ouvertes

Le domaine des méthodes formelles et des techniques de validation pour la sécurité des systèmes automobiles est en constante évolution, alimenté par la recherche continue et les avancées technologiques. Cette section propose une exploration approfondie des tendances actuelles de la recherche et expose les questions ouvertes cruciales qui nécessitent des investigations supplémentaires.

## 7.1 Revues bibliographiques

Une revue de littérature exhaustive révèle un paysage dynamique marqué par un regain d'intérêt pour l'intégration des méthodes formelles et des techniques de validation pour la sécurité des systèmes automobiles. Les chercheurs ont entrepris une diversité d'études, allant du développement de nouveaux modèles formels à l'exploration de méthodologies de validation innovantes. Des contributions remarquables ont abordé des défis critiques, tels que la scalabilité des processus de vérification, l'application de techniques mathématiques avancées et la proposition de cadres pour une intégration transparente dans le cycle de développement automobile.

De plus, les chercheurs ont fait des avancées significatives pour combler le fossé entre les méthodes formelles théoriques et leur mise en œuvre pratique, comme en témoignent des études de cas présentant des applications réussies dans des systèmes automobiles du monde réel. Ces études servent de témoignages convaincants de l'efficacité de ces méthodologies, offrant des aperçus précieux sur les défis et les solutions pratiques pour garantir la sécurité des systèmes automobiles.

## 7.2 Tendances émergentes

Le paysage en constante évolution des méthodes formelles et des techniques de validation pour la sécurité des systèmes automobiles est marqué par l'émergence de tendances innovantes qui façonnent l'avenir du domaine. Notamment, les chercheurs explorent l'intégration d'algorithmes d'apprentissage automatique comme moyen de renforcer les évaluations de sécurité. Les techniques d'apprentissage automatique ont le potentiel d'améliorer l'automatisation et la précision dans l'identification des vulnérabilités de sécurité, représentant un changement de paradigme dans l'évaluation de la sécurité des systèmes automobiles.

De plus, l'intégration de la technologie de la blockchain a suscité une attention considérable. Les caractéristiques intrinsèques de la blockchain, telles que la résistance à la falsification et les mécanismes de consensus décentralisés, offrent des opportunités pour garantir l'intégrité et la traçabilité des mises à jour logicielles. De plus, l'utilisation de contrats intelligents au sein des écosystèmes de la blockchain présente des promesses pour établir des accords sécurisés dans les interactions des systèmes automobiles.

## 7.3 Questions de recherche ouvertes

Malgré les progrès réalisés, des questions de recherche ouvertes cruciales persistent dans le domaine des méthodes formelles et des techniques de validation pour la sécurité des systèmes automobiles. Une préoccupation urgente concerne le développement de techniques capables de gérer efficacement les exigences de scalabilité croissantes posées par des systèmes automobiles de plus en plus complexes. À mesure que les véhicules deviennent plus connectés et autonomes, la complexité de la vérification de leurs propriétés de sécurité devient primordiale.

De plus, il existe un besoin pressant de méthodologies capables de s'adapter de manière transparente à l'évolution du paysage des menaces. Garantir la résilience face aux

cyberattaques sophistiquées reste un défi de taille, nécessitant des approches innovantes qui vont au-delà des paradigmes de sécurité conventionnels.

Une zone prometteuse pour les futures explorations se situe à l'intersection des méthodes formelles, de l'apprentissage automatique et de la technologie de la blockchain. Étudier comment ces technologies synergiques peuvent être exploitées pour renforcer les mesures de sécurité représente une frontière passionnante avec le potentiel de révolutionner la sécurité des systèmes automobiles.

En résumé, le paysage actuel de la recherche sur les méthodes formelles et les techniques de validation pour la sécurité des systèmes automobiles est marqué par une interaction dynamique entre les revues bibliographiques, les études de cas éclairantes, l'exploration des tendances émergentes et la recherche de réponses aux questions de recherche ouvertes. Ce domaine vibrant promet de façonner considérablement l'avenir des systèmes automobiles sécurisés, en les protégeant contre une gamme toujours croissante de menaces cybernétiques.

# 8 Conclusions

Dans la quête de la sécurité des systèmes automobiles, l'intégration des méthodes formelles et des techniques de validation se présente comme une approche redoutable. Cette étude a exploré le paysage de ces méthodologies, mettant en lumière leur application, leurs avantages et leurs limites. Dans cette section de conclusion, nous résumons les principales conclusions et discutons de leurs contributions et implications.

L'étude a mis en évidence le rôle central joué par les méthodes formelles et les techniques de validation dans la garantie de la sécurité des systèmes automobiles. Des premières étapes de l'ingénierie des exigences à la phase de test, ces méthodologies fournissent un moyen structuré et systématique d'identifier et de réduire les vulnérabilités de sécurité. Les méthodes formelles, telles que la vérification de modèles et la démonstration de théorèmes, offrent une rigueur mathématique pour vérifier les propriétés de sécurité, tandis que les techniques de validation, telles que les tests de pénétration et l'injection de fautes, fournissent une validation pratique des mesures de sécurité.

L'intégration de ces méthodologies au sein du cycle de développement automobile favorise une approche holistique de la sécurité, l'incorporant dans la conception et la mise en œuvre même du système. Les études de cas et les tendances émergentes soulignent en outre l'applicabilité pratique et la nature évolutive de ces techniques, mettant en évidence leur rôle essentiel dans la protection des systèmes automobiles.

Les contributions de cette étude résident dans la fourniture d'un aperçu complet des méthodes formelles et des techniques de validation de pointe pour la sécurité des systèmes automobiles. En synthétisant les résultats de recherche, les études de cas et les tendances émergentes, cette étude constitue une ressource précieuse pour les chercheurs, les praticiens et les décideurs impliqués dans la conception, le développement et l'évaluation des systèmes automobiles sécurisés.

De plus, les implications de cette étude s'étendent au paysage plus large de la cybersécurité. Les méthodologies discutées ici offrent des aperçus et des principes précieux qui sont transférables à d'autres domaines présentant des préoccupations similaires en matière de sécurité. L'accent mis sur les mesures de sécurité proactives, la vérification

rigoureuse et les scénarios de validation du monde réel fournit un modèle pour renforcer les systèmes face à un paysage de menaces en constante évolution.

En conclusion, l'intégration des méthodes formelles et des techniques de validation représente un paradigme crucial dans la sécurité des systèmes automobiles. Cette étude souligne leur importance, offrant une feuille de route pour renforcer les systèmes automobiles contre un éventail de menaces potentielles. À mesure que l'industrie automobile continue d'évoluer, ces méthodologies resteront essentielles pour garantir la sécurité et l'intégrité des véhicules de demain.

## Références


[1] Information system security assessment framework (issaf). `https://www.futurelearn.com/info/courses/ethical-hacking-an-introduction/0/steps/71521`. (Accessed on 11/08/2023).

[2] Nist sp 800-115 | nist. `https://www.nist.gov/privacy-framework/nist-sp-800-115`. (Accessed on 11/08/2023).

[3] The penetration testing execution standard. `http://www.pentest-standard.org/index.php/Main_Page`. (Accessed on 11/08/2023).

[4] Sa'ed Abed, Adnan Rashid, and Osman Hasan. Formal analysis of unmanned aerial vehicles using higher-order-logic theorem proving. *Journal of Aerospace Information Systems*, 17(9) :481–495, 2020.

[5] Emad H Abualsauod. A hybrid blockchain method in internet of things for privacy and security in unmanned aerial vehicles network. *Computers and Electrical Engineering*, 99 :107847, 2022.

[6] Mazen Ahmed and Mona Safar. Symbolic execution based verification of compliance with the iso 26262 functional safety standard. In *2019 14th International Conference on Design & Technology of Integrated Systems In Nanoscale Era (DTIS)*, pages 1–6. IEEE, 2019.

[7] Mahmoud Alfadel, Nicholas Alexandre Nagy, Diego Elias Costa, Rabe Abdalkareem, and Emad Shihab. Empirical analysis of security-related code reviews in npm packages. *Journal of Systems and Software*, 203 :111752, 2023.

[8] Elmustafa Sayed Ali, Mohammad Kamrul Hasan, Rosilah Hassan, Rashid A Saeed, Mona Bakri Hassan, Shayla Islam, Nazmus Shaker Nafi, and Savitri Bevinakoppa. Machine learning technologies for secure vehicular communication in internet of vehicles : recent advances and applications. *Security and Communication Networks*, 2021 :1–23, 2021.

[9] Omar Azib Alkhudaydi, Moez Krichen, and Ans D Alghamdi. A deep learning methodology for predicting cybersecurity attacks on the internet of things. *Information*, 14(10) :550, 2023.

[10] Sadeq Almeaibed, Saba Al-Rubaye, Antonios Tsourdos, and Nicolas P Avdelidis. Digital twin analysis to promote safety and security in autonomous vehicles. *IEEE Communications Standards Magazine*, 5(1) :40–46, 2021.

[11] Mohammed Alshahrani and Issa Traore. Secure mutual authentication and automated access control for iot smart home using cumulative keyed-hash chain. *Journal of information security and applications*, 45 :156–175, 2019.

[12] Hamoud Alshammari, Karim Gasmi, Ibtihel Ben Ltaifa, Moez Krichen, Lassaad Ben Ammar, and Mahmood A Mahmood. Olive disease classification based on vision transformer and cnn models. *Computational Intelligence and Neuroscience*, 2022, 2022.

[13] Esra Abdullatif Altulaihan, Abrar Alismail, and Mounir Frikha. A survey on web application penetration testing. *Electronics*, 12(5) :1229, 2023.

[14] Md Tarique Jamal Ansari, Dhirendra Pandey, and Mamdouh Alenezi. Store : Security threat oriented requirements engineering methodology. *Journal of King Saud University-Computer and Information Sciences*, 34(2) :191–203, 2022.

[15] Hala Assal. Collaborative security code review. In *Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia*, pages 439–444, 2015.

[16] Omid Avatefipour, Ameena Saad Al-Sumaiti, Ahmed M El-Sherbeeny, Emad Mahrous Awwad, Mohammed A Elmeligy, Mohamed A Mohamed, and Hafiz Malik. An intelligent secured framework for cyberattack detection in electric vehicles' can bus using machine learning. *IEEE Access*, 7 :127580–127592, 2019.

[17] Rubby Aworka, Lontsi Saadio Cedric, Wilfried Yves Hamilton Adoni, Jérémie Thouakesseh Zoueu, Franck Kalala Mutombo, Charles Lebon Mberi Kimpolo, Tarik Nahhal, and Moez Krichen. Agricultural decision system based on advanced machine learning models for yield prediction : Case of east african countries. *Smart Agricultural Technology*, 2 :100048, 2022.

[18] Murat Aydos, Çiğdem Aldan, Evren Coşkun, and Alperen Soydan. Security testing of web applications : A systematic mapping of the literature. *Journal of King Saud University-Computer and Information Sciences*, 34(9) :6775–6792, 2022.

[19] Roberto Baldoni, Emilio Coppa, Daniele Cono D'elia, Camil Demetrescu, and Irene Finocchi. A survey of symbolic execution techniques. *ACM Computing Surveys (CSUR)*, 51(3) :1–39, 2018.

[20] Vitor Bandeira, Felipe Rosa, Ricardo Reis, and Luciano Ost. Non-intrusive fault injection techniques for efficient soft error vulnerability analysis. In *2019 IFIP/IEEE 27th International Conference on Very Large Scale Integration (VLSI-SoC)*, pages 123–128. IEEE, 2019.

[21] Abdelhakim Baouya, Otmane Ait Mohamed, Samir Ouchani, and Djamal Bennouar. Reliability-driven automotive software deployment based on a parametrizable probabilistic model checking. *Expert Systems with Applications*, 174 :114572, 2021.

[22] Martín Barrère, Chris Hankin, Nicolas Nicolaou, Demetrios G Eliades, and Thomas Parisini. Measuring cyber-physical security in industrial control systems via minimum-effort attack strategies. *Journal of information security and applications*, 52 :102471, 2020.

[23] Ryan Beckett, Aarti Gupta, Ratul Mahajan, and David Walker. Abstract interpretation of distributed network control planes. *Proceedings of the ACM on Programming Languages*, 4(POPL) :1–27, 2019.

[24] Shanay Behrad, Emmanuel Bertin, Stéphane Tuffin, and Noel Crespi. A new scalable authentication and access control mechanism for 5g-based iot. *Future Generation Computer Systems*, 108 :46–61, 2020.

[25] Moritz Beller, Radjino Bholanath, Shane McIntosh, and Andy Zaidman. Analyzing the state of static analysis : A large-scale evaluation in open source software. In *2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, volume 1, pages 470–481. IEEE, 2016.

[26] Gueltoum Bendiab, Amina Hameurlaine, Georgios Germanos, Nicholas Kolokotronis, and Stavros Shiaeles. Autonomous vehicles security : Challenges and solutions using blockchain and artificial intelligence. *IEEE Transactions on Intelligent Transportation Systems*, 2023.

[27] Samaresh Bera, Sudip Misra, and Athanasios V Vasilakos. Software-defined networking for internet of things : A survey. *IEEE Internet of Things Journal*, 4(6) :1994–2008, 2017.

[28] Béatrice Bérard, Michel Bidoit, Alain Finkel, François Laroussinie, Antoine Petit, Laure Petrucci, and Philippe Schnoebelen. *Systems and software verification : model-checking techniques and tools*. Springer Science & Business Media, 2013.

[29] Hunter Berry, Mai A Abdel-Malek, and Ahmed S Ibrahim. A machine learning approach for combating cyber attacks in self-driving vehicles. In *SoutheastCon 2021*, pages 1–3. IEEE, 2021.

[30] Wolfgang Bibel. *Automated theorem proving*. Springer Science & Business Media, 2013.

[31] Wadii Boulila, Maha Driss, Eman Alshanqiti, Mohamed Al-Sarem, Faisal Saeed, and Moez Krichen. Weight initialization techniques for deep learning algorithms in remote sensing : Recent trends and future perspectives. *Advances on Smart and Soft Computing : Proceedings of ICACIn 2021*, pages 477–484, 2022.

[32] Zakaria Boulouard, Mariyam Ouaissa, Mariya Ouaissa, Farhan Siddiqui, Mutiq Almutiq, and Moez Krichen. An integrated artificial intelligence of things environment for river flood prevention. *Sensors*, 22(23) :9485, 2022.

[33] Guillaume Brat, Jorge A Navas, Nija Shi, and Arnaud Venet. Ikos : A framework for static analysis based on abstract interpretation. In *Software Engineering and Formal Methods : 12th International Conference, SEFM 2014, Grenoble, France, September 1-5, 2014. Proceedings 12*, pages 271–277. Springer, 2014.

[34] Larissa Braz and Alberto Bacchelli. Software security during modern code review : the developer's perspective. In *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 810–821, 2022.

[35] Holger Breuing, Lucas Heil, and Bernd Vierling. It security for the entire automotive ecosystem. *ATZelectronics worldwide*, 14(7) :60–63, 2019.

[36] Andrew Buttner, Richard Piazza, Rushi Purohit, and Alec Summers. A secure code review retrospective. In *2020 IEEE Secure Development (SecDev)*, pages 31–32. IEEE, 2020.

[37] Nelson H Carreras Guzman, Morten Wied, Igor Kozine, and Mary Ann Lundteigen. Conceptualizing the key features of cyber-physical systems in a multi-layered representation for safety and security analysis. *Systems Engineering*, 23(2) :189–210, 2020.

[38] Lontsi Saadio Cedric, Wilfried Yves Hamilton Adoni, Rubby Aworka, Jérémie Thouakesseh Zoueu, Franck Kalala Mutombo, Moez Krichen, and Charles Lebon Mberi Kimpolo. Crops yield prediction based on machine learning models : Case of west african countries. *Smart Agricultural Technology*, 2 :100049, 2022.

[39] Haoye Chai, Supeng Leng, Yijin Chen, and Ke Zhang. A hierarchical blockchain-enabled federated learning algorithm for knowledge sharing in internet of vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 22(7) :3975–3986, 2020.

[40] Ursula Challita, Aidin Ferdowsi, Mingzhe Chen, and Walid Saad. Machine learning for wireless connectivity and security of cellular-connected uavs. *IEEE Wireless Communications*, 26(1) :28–35, 2019.

[41] Chen Chen, Tingting Xiao, Tie Qiu, Ning Lv, and Qingqi Pei. Smart-contract-based economical platooning in blockchain-enabled urban internet of vehicles. *IEEE Transactions on Industrial Informatics*, 16(6) :4122–4133, 2019.

[42] Yunja Choi. Model checking trampoline os : a case study on safety analysis for automotive software. *Software Testing, Verification and Reliability*, 24(1) :38–60, 2014.

[43] Edmund M Clarke, Thomas A Henzinger, Helmut Veith, Roderick Bloem, et al. *Handbook of model checking*, volume 10. Springer, 2018.

[44] Stephen A Cook. The complexity of theorem-proving procedures. In *Logic, Automata, and Computational Complexity : The Works of Stephen A. Cook*, pages 143–152. 2023.

[45] Domenico Cotroneo, Luigi De Simone, and Roberto Natella. Thorfi : a novel approach for network fault injection as a service. *Journal of Network and Computer Applications*, 201 :103334, 2022.

[46] Patrick Cousot. *Principles of Abstract Interpretation*. MIT Press, 2021.

[47] Patrick Cousot and Radhia Cousot. Abstract interpretation : past, present and future. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–10, 2014.

[48] Patrick Cousot and Michael Monerau. Probabilistic abstract interpretation. In *European Symposium on Programming*, pages 169–193. Springer, 2012.

[49] Cas Cremers, Charlie Jacomme, and Philip Lukert. Subterm-based proof techniques for improving the automation and scope of security protocol analysis. In

*2023 IEEE 36th Computer Security Foundations Symposium (CSF)*, pages 200–213. IEEE, 2023.

[50] Venia Noella Nanisura Damanik and Septia Ulfa Sunaringtyas. Secure code recommendation based on code review result using owasp code review guide. In *2020 International Workshop on Big Data and Information Security (IWBIS)*, pages 153–158. IEEE, 2020.

[51] Seyyed Jalaladdin Hosseini Dehshiri, Mir Seyed Mohammad Mohsen Emamat, and Maghsoud Amiri. A novel group bwm approach to evaluate the implementation criteria of blockchain technology in the automotive industry supply chain. *Expert Systems with Applications*, 198 :116826, 2022.

[52] R Dhaya, R Kanthavel, and Kanagaraj Venusamy. Cloud computing security protocol analysis with parity-based distributed file system. *Annals of Operations Research*, pages 1–20, 2021.

[53] Marco di Biase, Magiel Bruntink, and Alberto Bacchelli. A security perspective on code review : The case of chromium. In *2016 IEEE 16th International Working Conference on Source Code Analysis and Manipulation (SCAM)*, pages 21–30. IEEE, 2016.

[54] Jürgen Dobaj, Georg Macher, Damjan Ekert, Andreas Riel, and Richard Messnarz. Towards a security-driven automotive development lifecycle. *Journal of Software : Evolution and Process*, 35(8) :e2407, 2023.

[55] Andrea Domenici, Adriano Fagiolini, and Maurizio Palmieri. Integrated simulation and formal verification of a simple autonomous vehicle. In *Software Engineering and Formal Methods : SEFM 2017 Collocated Workshops : DataMod, FAACS, MSE, CoSim-CPS, and FOCLASA, Trento, Italy, September 4-5, 2017, Revised Selected Papers 15*, pages 300–314. Springer, 2018.

[56] Wei Dong, Tingting Wang, Liang Zhang, and Hao Fan. Security protocol analysis based on run modes and petri net. In *International Conference on Algorithms, Microchips and Network Applications*, volume 12176, pages 397–401. SPIE, 2022.

[57] Sri Yogesh Dorbala and Robin Singh Bhadoria. Analysis for security attacks in cyber-physical systems. *Cyber-Physical Systems : A Computational Perspective*, pages 395–414, 2015.

[58] Ali Dorri, Marco Steger, Salil S Kanhere, and Raja Jurdak. Blockchain : A distributed solution to automotive security and privacy. *IEEE Communications Magazine*, 55(12) :119–125, 2017.

[59] Dan Dragomir, Laura Gheorghe, Sergiu Costea, and Alexandru Radovici. A survey on secure communication protocols for iot systems. In *2016 international workshop on Secure Internet of Things (SIoT)*, pages 47–62. IEEE, 2016.

[60] Christof Ebert and Ruschil Ray. Penetration testing for automotive cybersecurity. *ATZelectronics worldwide*, 16(6) :16–22, 2021.

[61] Rayane El Sibai, Nader Gemayel, Jacques Bou Abdo, and Jacques Demerjian. A survey on access control mechanisms for cloud computing. *Transactions on Emerging Telecommunications Technologies*, 31(2) :e3720, 2020.

[62] Mohammad Eslami, Behnam Ghavami, Mohsen Raji, and Ali Mahani. A survey on fault injection methods of digital integrated circuits. *Integration*, 71 :154–163, 2020.

[63] Manuel Fähndrich and Francesco Logozzo. Static contract checking with abstract interpretation. In *International conference on formal verification of object-oriented software*, pages 10–30. Springer, 2010.

[64] Thomas Faschang and Georg Macher. An open software-based framework for automotive cybersecurity testing. In *European Conference on Software Process Improvement*, pages 316–328. Springer, 2023.

[65] Aidin Ferdowsi, Ursula Challita, Walid Saad, and Narayan B Mandayam. Robust deep reinforcement learning for security and safety in autonomous vehicle systems. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 307–312. IEEE, 2018.

[66] Luca Ferretti, Mirco Marchetti, and Michele Colajanni. Fog-based secure communications for low-power iot devices. *ACM Transactions on Internet Technology (TOIT)*, 19(2) :1–21, 2019.

[67] Eric Filiol, Francesco Mercaldo, and Antonella Santone. A method for automatic penetration testing and mitigation : A red hat approach. *Procedia Computer Science*, 192 :2039–2046, 2021.

[68] Institute for Security and Open Methodologies. The Open Source Security Testing Methodology Manual. `https://www.isecom.org/OSSTMM.3.pdf`, 2010. [Accessed 08-11-2023].

[69] Daniel S Fowler, Jeremy Bryans, Madeline Cheah, Paul Wooderson, and Siraj A Shaikh. A method for constructing automotive cybersecurity tests, a can fuzz testing example. In *2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pages 1–8. IEEE, 2019.

[70] Daniel S Fowler, Jeremy Bryans, Siraj Ahmed Shaikh, and Paul Wooderson. Fuzz testing for automotive cyber-security. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 239–246. IEEE, 2018.

[71] Paula Fraga-Lamas and Tiago M Fernández-Caramés. A review on blockchain technologies for an advanced and cyber-resilient automotive industry. *IEEE access*, 7 :17578–17598, 2019.

[72] Daniel J Fremont, Edward Kim, Yash Vardhan Pant, Sanjit A Seshia, Atul Acharya, Xantha Bruso, Paul Wells, Steve Lemke, Qiang Lu, and Shalin Mehta. Formal scenario-based testing of autonomous vehicles : From simulation to the real world. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2020.

[73] Vaishnavi Ganesh and Manmohan Sharma. Intrusion detection and prevention systems : A review. *Inventive Communication and Computational Technologies : Proceedings of ICICCT 2020*, pages 835–844, 2021.

[74] Aakash Gangolli, Qusay H Mahmoud, and Akramul Azim. A systematic review of fault injection attacks on iot systems. *Electronics*, 11(13) :2023, 2022.

[75] Tiago Espinha Gasiba, Ulrike Lechner, Maria Pinto-Albuquerque, and Daniel Mendez Fernandez. Awareness of secure coding guidelines in the industry-a first data analysis. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 345–352. IEEE, 2020.

[76] Roberto Giacobazzi and Francesco Ranzato. History of abstract interpretation. *IEEE Annals of the History of Computing*, 44(2) :33–43, 2021.

[77] Thomas Given-Wilson, Nisrine Jafri, and Axel Legay. Combined software and hardware fault injection vulnerability detection. *Innovations in Systems and Software Engineering*, 16 :101–120, 2020.

[78] Vibhav Gogate and Pedro Domingos. Probabilistic theorem proving. *Communications of the ACM*, 59(7) :107–115, 2016.

[79] Paulin Gohoungodji, Amoin Bernadine N'Dri, Jean-Michel Latulippe, and Adriana Leiria Barreto Matos. What is stopping the automotive industry from going green ? a systematic review of barriers to green innovation in the automotive industry. *Journal of Cleaner Production*, 277 :123524, 2020.

[80] Meriem Guerar, Luca Verderame, Alessio Merlo, Francesco Palmieri, Mauro Migliardi, and Luca Vallerini. Circlepin : a novel authentication mechanism for smartwatches to prevent unauthorized access to iot devices. *ACM Transactions on Cyber-Physical Systems*, 4(3) :1–19, 2020.

[81] Shengjian Guo, Meng Wu, and Chao Wang. Symbolic execution of programmable logic controller code. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, pages 326–336, 2017.

[82] Sohan Gyawali and Yi Qian. Misbehavior detection using machine learning in vehicular communication networks. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2019.

[83] Jon Duncan Hagar. Security owasp iot information pointer and logging events. In *IoT System Testing : An IoT Journey from Devices to Analytics and the Edge*, pages 209–215. Springer, 2022.

[84] Subir Halder, Amrita Ghosal, and Mauro Conti. Secure over-the-air software updates in connected vehicles : A survey. *Computer Networks*, 178 :107343, 2020.

[85] Jia Cheng Han and Zhi Quan Zhou. Metamorphic fuzz testing of autonomous vehicles. In *Proceedings of the IEEE/ACM 42nd International Conference on Software Engineering Workshops*, pages 380–385, 2020.

[86] Zhaoyang Han, Yaoqi Yang, Muhammad Bilal, Weizheng Wang, Moez Krichen, Abeer Abdullah Alsadhan, and Chunpeng Ge. Smart optimization solution for channel access attack defense under uav-aided heterogeneous network. *IEEE Internet of Things Journal*, 2023.

[87] Anand Handa, Ashu Sharma, and Sandeep K Shukla. Machine learning in cyber-

security : A review. *Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery*, 9(4) :e1306, 2019.

[88] Hazim Hanif, Mohd Hairul Nizam Md Nasir, Mohd Faizal Ab Razak, Ahmad Firdaus, and Nor Badrul Anuar. The rise of software vulnerability : Taxonomy of software vulnerabilities detection and machine learning approaches. *Journal of Network and Computer Applications*, 179 :103009, 2021.

[89] Jingjing Hao and Guangsheng Han. On the modeling of automotive security : A survey of methods and perspectives. *Future Internet*, 12(11) :198, 2020.

[90] John Harrison. *Theorem proving with the real numbers*. Springer Science & Business Media, 2012.

[91] John Harrison, Josef Urban, and Freek Wiedijk. History of interactive theorem proving. In *Computational Logic*, volume 9, pages 135–214, 2014.

[92] Amal Hbaieb, Samiha Ayed, and Lamia Chaari. A survey of trust management in the internet of vehicles. *Computer Networks*, 203 :108558, 2022.

[93] Arthur Hicken. Mitigate risk with leveraging automotive development standards. *ATZelektronik worldwide*, 13(1) :42–47, 2018.

[94] Jun Huang, Mingli Zhao, Yide Zhou, and Cong-Cong Xing. In-vehicle networking : Protocols, challenges, and solutions. *IEEE Network*, 33(1) :92–98, 2018.

[95] Xumin Huang, Dongdong Ye, Rong Yu, and Lei Shu. Securing parked vehicle assisted fog computing with blockchain and optimal smart contract design. *IEEE/CAA Journal of Automatica Sinica*, 7(2) :426–441, 2020.

[96] Rateb Jabbar. Noora fetais, mohamed kharbeche, moez krichen, kamel barkaoui, and mohammed shinoy. blockchain for the internet of vehicles : How to use blockchain to secure vehicle-to-everything (v2x) communication and payment. *IEEE Sensors Journal*, 21(14) :15807–15823, 2021.

[97] Rateb Jabbar, Moez Krichen, Noora Fetais, and Kamel Barkaoui. Adopting formal verification and model-based testing techniques for validating a blockchain-based healthcare records sharing system. In *22nd International Conference on Enterprise Information Systems*, pages 261–268. SCITEPRESS-Science and Technology Publications, 2020.

[98] Rateb Jabbar, Moez Krichen, Mohamed Kharbeche, Noora Fetais, and Kamel Barkaoui. A formal model-based testing framework for validating an iot solution for blockchain-based vehicles communication. In *15th International Conference on Evaluation of Novel Approaches to Software Engineering*, pages 595–602. SCITEPRESS-Science and Technology Publications, 2020.

[99] Rateb Jabbar, Moez Krichen, Mohammed Shinoy, Mohamed Kharbeche, Noora Fetais, and Kamel Barkaoui. A model-based and resource-aware testing framework for parking system payment using blockchain. In *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pages 1252–1259. IEEE, 2020.

[100] Milena Vujošević Janičić, Ognjen Plavšić, Mirko Brkušanin, and Petar Jovanović. Autocheck : A tool for checking compliance with automotive coding standards. In

*2021 Zooming Innovation in Consumer Technologies Conference (ZINC)*, pages 150–155. IEEE, 2021.

[101] Uzair Javaid, Muhammad Naveed Aman, and Biplab Sikdar. Drivman : Driving trust management and data sharing in vanets with blockchain and smart contracts. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pages 1–5. IEEE, 2019.

[102] Bryer Jeannotte and Ali Tekeoglu. Artorias : Iot security testing framework. In *2019 26th International Conference on Telecommunications (ICT)*, pages 233–237. IEEE, 2019.

[103] Saurabh Jha, Subho Banerjee, Timothy Tsai, Siva KS Hari, Michael B Sullivan, Zbigniew T Kalbarczyk, Stephen W Keckler, and Ravishankar K Iyer. Ml-based fault injection for autonomous vehicles : A case for bayesian fault injection. In *2019 49th annual IEEE/IFIP international conference on dependable systems and networks (DSN)*, pages 112–124. IEEE, 2019.

[104] Rahul Johari, Ishveen Kaur, Reena Tripathi, and Kanika Gupta. Penetration testing in iot network. In *2020 5th International Conference on Computing, Communication and Security (ICCCS)*, pages 1–7. IEEE, 2020.

[105] Pontus Johnson, Robert Lagerström, and Mathias Ekstedt. A meta language for threat modeling and attack simulations. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, pages 1–8, 2018.

[106] Daniel Kaestner, Bernard Schmidt, Maximilian Schlund, Laurent Mauborgne, Stephan Wilhelm, and Christian Ferdinand. Analyze this ! sound static analysis for integration verification of large-scale automotive software. Technical report, SAE Technical Paper, 2019.

[107] Cezary Kaliszyk and Josef Urban. Learning-assisted theorem proving with millions of lemmas. *Journal of symbolic computation*, 69 :109–128, 2015.

[108] Aikaterini Kanta, Iwen Coisel, and Mark Scanlon. A survey exploring open source intelligence for smarter password cracking. *Forensic Science International : Digital Investigation*, 35 :301075, 2020.

[109] Aikaterini Kanta, Iwen Coisel, and Mark Scanlon. Pcwq : A framework for evaluating password cracking wordlist quality. In *International Conference on Digital Forensics and Cyber Crime*, pages 159–175. Springer, 2021.

[110] Adi Karahasanovic, Pierre Kleberger, and Magnus Almgren. Adapting threat modeling methods for the automotive industry. In *Proceedings of the 15th ESCAR Conference*, pages 1–10, 2017.

[111] Abid Khan, Awais Ahmad, Mansoor Ahmed, Jadran Sessa, and Marco Anisetti. Authorization schemes for internet of things : requirements, weaknesses, future challenges and trends. *Complex & Intelligent Systems*, 8(5) :3919–3941, 2022.

[112] Navid Ali Khan, Noor Zaman Jhanjhi, Sarfraz Nawaz Brohi, and Anand Nayyar. Emerging use of uav's : secure communication protocol issues and challenges. In *Drones in smart-cities*, pages 37–55. Elsevier, 2020.

[113] Rafiullah Khan, Kieran McLaughlin, David Laverty, and Sakir Sezer. Stride-

based threat modeling for cyber-physical systems. In *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, pages 1–6. IEEE, 2017.

[114] Yugansh Khera, Deepansh Kumar, Nidhi Garg, et al. Analysis and impact of vulnerability assessment and penetration testing. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pages 525–530. IEEE, 2019.

[115] Aziz Altaf Khuwaja, Yunfei Chen, Nan Zhao, Mohamed-Slim Alouini, and Paul Dobbins. A survey of channel modeling for uav communications. *IEEE Communications Surveys & Tutorials*, 20(4) :2804–2821, 2018.

[116] Kyounggon Kim, Jun Seok Kim, Seonghoon Jeong, Jo-Hee Park, and Huy Kang Kim. Cybersecurity for autonomous vehicles : Review of attacks and defense. *Computers & Security*, 103 :102150, 2021.

[117] Yunho Kim, Dongju Lee, Junki Baek, and Moonzoo Kim. Maestro : Automated test generation framework for high test coverage and reduced human effort in automotive industry. *Information and Software Technology*, 123 :106221, 2020.

[118] Rhys Kirk, Hoang Nga Nguyen, Jeremy Bryans, Siraj Ahmed Shaikh, and Charles Wartnaby. A formal framework for security testing of automotive over-the-air update systems. *Journal of Logical and Algebraic Methods in Programming*, 130 :100812, 2023.

[119] George Klees, Andrew Ruef, Benji Cooper, Shiyi Wei, and Michael Hicks. Evaluating fuzz testing. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 2123–2138, 2018.

[120] Laura Kovács and Andrei Voronkov. First-order theorem proving and vampire. In *International Conference on Computer Aided Verification*, pages 1–35. Springer, 2013.

[121] Moez Krichen. *Model-based testing for real-time systems*. PhD thesis, PhD thesis, PhD thesis, Universit Joseph Fourier (December 2007), 2007.

[122] Moez Krichen. A formal framework for conformance testing of distributed real-time systems. In *International Conference On Principles Of Distributed Systems*, pages 139–142. Springer, 2010.

[123] Moez Krichen. A formal framework for black-box conformance testing of distributed real-time systems. *International Journal of Critical Computer-Based Systems*, 3(1-2) :26–43, 2012.

[124] Moez Krichen. *Contributions to model-based testing of dynamic and distributed real-time systems*. PhD thesis, École Nationale d'Ingénieurs de Sfax (Tunisie), 2018.

[125] Moez Krichen. Anomalies detection through smartphone sensors : A review. *IEEE Sensors Journal*, 21(6) :7207–7217, 2021.

[126] Moez Krichen. How artificial intelligence can revolutionize software testing techniques. In *International Conference on Innovations in Bio-Inspired Computing and Applications*, pages 189–198. Springer Nature Switzerland Cham, 2022.

[127] Moez Krichen. Convolutional neural networks : A survey. *Computers*, 12(8) :151, 2023.

[128] Moez Krichen. Deep reinforcement learning. In *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–7. IEEE, 2023.

[129] Moez Krichen. Formal methods and validation techniques for ensuring automotive systems security. *Information*, 14(12) :666, 2023.

[130] Moez Krichen. Generative adversarial networks. In *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–7. IEEE, 2023.

[131] Moez Krichen. Strengthening the security of smart contracts through the power of artificial intelligence. *Computers*, 12(5) :107, 2023.

[132] Moez Krichen. A survey on formal verification and validation techniques for internet of things. *Applied Sciences*, 13(14) :8122, 2023.

[133] Moez Krichen, Wilfried Yves Hamilton Adoni, Alaeddine Mihoub, Mohammed Y Alzahrani, and Tarik Nahhal. Security challenges for drone communications : Possible threats, attacks and countermeasures. In *2022 2nd International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, pages 184–189. IEEE, 2022.

[134] Moez Krichen and Roobaea Alroobaea. A new model-based framework for testing security of iot systems in smart cities using attack trees and price timed automata. In *14th International Conference on Evaluation of Novel Approaches to Software Engineering - ENASE 2019*, 2019.

[135] Moez Krichen, Meryem Ammi, Alaeddine Mihoub, and Qasem Abu Al-Haija. Short survey on using blockchain technology in modern wireless networks, iot and smart grids. In *International Conference on Cybersecurity, Cybercrimes, and Smart Emerging Technologies*, pages 163–173. Springer International Publishing Cham, 2022.

[136] Moez Krichen, Omar Cheikhrouhou, Mariam Lahami, Roobaea Alroobaea, and Afef Jmal Maâlej. Towards a model-based testing framework for the security of internet of things for smart city applications. In *Smart Societies, Infrastructure, Technologies and Applications : First International Conference, SCITA 2017, Jeddah, Saudi Arabia, November 27–29, 2017, Proceedings 1*, pages 360–365. Springer International Publishing, 2018.

[137] Moez Krichen, Mariam Lahami, and Qasem Abu Al-Haija. Formal methods for the verification of smart contracts : A review. In *2022 15th International Conference on Security of Information and Networks (SIN)*, pages 01–08. IEEE, 2022.

[138] Moez Krichen, Mariam Lahami, and Qasem Abu Al-Haija. Formal methods for the verification of smart contracts : A review. In *2022 15th International Conference on Security of Information and Networks (SIN)*, pages 01–08. IEEE, 2022.

[139] Moez Krichen, Mariam Lahami, Omar Cheikhrouhou, Roobaea Alroobaea, and Afef Jmal Maâlej. Security testing of internet of things for smart city applications : A formal approach. In *Smart Infrastructure and Applications*, pages 629–653. Springer, Cham, 2020.

[140] Moez Krichen, Afef Jmal Maâlej, and Mariam Lahami. A model-based approach to combine conformance and load tests : an ehealth case study. *International Journal of Critical Computer-Based Systems*, 8(3-4) :282–310, 2018.

[141] Moez Krichen, Seifeddine Mechti, Roobaea Alroobaea, Elyes Said, Parminder Singh, Osamah Ibrahim Khalaf, and Mehedi Masud. A formal testing model for operating room control system using internet of things. *Computers, Materials & Continua*, 66(3) :2997–3011, 2021.

[142] Moez Krichen and Alaeddine Mihoub. Unmanned aerial vehicles communications security challenges : A survey. In *Unmanned Aerial Vehicles Applications : Challenges and Trends*, pages 349–373. Springer International Publishing Cham, 2023.

[143] Moez Krichen and Stavros Tripakis. Real-time testing with timed automata testers and coverage criteria. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, pages 134–151. Springer, Berlin, Heidelberg, 2004.

[144] Moez Krichen and Stavros Tripakis. State identification problems for timed automata. In *Testing of Communicating Systems : 17th IFIP TC6/WG 6.1 International Conference, TestCom 2005, Montreal, Canada, May 31-June, 2005. Proceedings 17*, pages 175–191. Springer Berlin Heidelberg, 2005.

[145] Moez Krichen and Stavros Tripakis. Interesting properties of the real-time conformance relation tioco. In *International Colloquium on Theoretical Aspects of Computing*, pages 317–331. Springer Berlin Heidelberg Berlin, Heidelberg, 2006.

[146] Elson Kurian, Daniela Briola, Pietro Braione, and Giovanni Denaro. Automatically generating test cases for safety-critical software via symbolic execution. *Journal of Systems and Software*, 199 :111629, 2023.

[147] Mariam Lahami, Fairouz Fakhfakh, Moez Krichen, and Mohamed Jmaiel. Towards a ttcn-3 test system for runtime testing of adaptable and distributed systems. In *Testing Software and Systems : 24th IFIP WG 6.1 International Conference, ICTSS 2012, Aalborg, Denmark, November 19-21, 2012. Proceedings 24*, pages 71–86. Springer Berlin Heidelberg, 2012.

[148] Mariam Lahami, Moez Krichen, Hajer Barhoumi, and Mohamed Jmaiel. Selective test generation approach for testing dynamic behavioral adaptations. In *Testing Software and Systems : 27th IFIP WG 6.1 International Conference, ICTSS 2015, Sharjah and Dubai, United Arab Emirates, November 23-25, 2015, Proceedings 27*, pages 224–239. Springer International Publishing, 2015.

[149] Mariam Lahami, Moez Krichen, Mariam Bouchakwa, and Mohamed Jmaiel. Using knapsack problem model to design a resource aware test architecture for adaptable and distributed systems. In *Testing Software and Systems : 24th IFIP WG 6.1 International Conference, ICTSS 2012, Aalborg, Denmark, November*

*19-21, 2012. Proceedings 24*, pages 103–118. Springer Berlin Heidelberg, 2012.

[150] Mariam Lahami, Moez Krichen, and Mohamed Jmaïel. Runtime testing approach of structural adaptations for dynamic and distributed systems. *International Journal of Computer Applications in Technology*, 51(4) :259–272, 2015.

[151] Mariam Lahami, Afef Jmal Maâlej, Moez Krichen, and Mohamed Amin Hammami. A comprehensive review of testing blockchain oriented software. *ENASE*, 182 :355–62, 2022.

[152] Ayyoub Lamssaggad, Nabil Benamar, Abdelhakim Senhaji Hafid, and Mounira Msahli. A survey on the current security landscape of intelligent transportation systems. *IEEE Access*, 9 :9180–9208, 2021.

[153] Timm Lauser, Daniel Zelle, and Christoph Krauß. Security analysis of automotive protocols. In *Proceedings of the 4th ACM Computer Science in Cars Symposium*, pages 1–12, 2020.

[154] Axel Legay, Benoît Delahaye, and Saddek Bensalem. Statistical model checking : An overview. In *International conference on runtime verification*, pages 122–135. Springer, 2010.

[155] Caroline Lemieux and Koushik Sen. Fairfuzz : A targeted mutation strategy for increasing greybox fuzz testing coverage. In *Proceedings of the 33rd ACM/IEEE international conference on automated software engineering*, pages 475–485, 2018.

[156] Teri Lenard and Roland Bolboaca. A statefull firewall and intrusion detection system enforced with secure logging for controller area network. In *European Interdisciplinary Cybersecurity Conference*, pages 39–45, 2021.

[157] Jie Liang, Mingzhe Wang, Yuanliang Chen, Yu Jiang, and Renwei Zhang. Fuzz testing in practice : Obstacles and solutions. In *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 562–566. IEEE, 2018.

[158] Qin Lin, Stefan Mitsch, André Platzer, and John M Dolan. Safe and resilient practical waypoint-following for autonomous vehicles. *IEEE Control Systems Letters*, 6 :1574–1579, 2021.

[159] Haiqing Liu, Yan Zhang, Shiqiang Zheng, and Yuancheng Li. Electric vehicle power trading mechanism based on blockchain and smart contract in v2g network. *IEEE Access*, 7 :160546–160558, 2019.

[160] Xiao Liu, Xiaoting Li, Rupesh Prajapati, and Dinghao Wu. Deepfuzz : Automatic generation of syntax valid c programs for fuzz testing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1044–1051, 2019.

[161] Elio Jordan Lopes, Shaolin Kataria, Shashank Keshav, Sumaiya Thaseen Ikram, Muhammad Rukunuddin Ghalib, Achyut Shankar, and Moez Krichen. Live video streaming service with pay-as-you-use model on ethereum blockchain and interplanetary file system. *Wireless Networks*, 28(7) :3111–3125, 2022.

[162] Tamara Lopez, Helen Sharp, Thein Tun, Arosha Bandara, Mark Levine, and Bashar Nuseibeh. " hopefully we are mostly secure" : Views on secure code in professional practice. In *2019 IEEE/ACM 12th International Workshop on*

*Cooperative and Human Aspects of Software Engineering (CHASE)*, pages 61–68. IEEE, 2019.

[163] Donald W Loveland. *Automated theorem proving : A logical basis*. Elsevier, 2016.

[164] Feng Luo, Yifan Jiang, Zhaojing Zhang, Yi Ren, and Shuo Hou. Threat analysis and risk assessment for connected vehicles : A survey. *Security and Communication Networks*, 2021 :1–19, 2021.

[165] Feng Luo, Xuan Zhang, Zhenyu Yang, Yifan Jiang, Jiajia Wang, Mingzhi Wu, and Wanqiang Feng. Cybersecurity testing for automotive domain : A survey. *Sensors*, 22(23) :9211, 2022.

[166] Afef Jmal Maâlej, Manel Hamza, Moez Krichen, and Mohamed Jmaiel. Automated significant load testing for ws-bpel compositions. In *2013 IEEE sixth international conference on software testing, verification and validation workshops*, pages 144–153. IEEE, 2013.

[167] Afef Jmal Maâlej and Moez Krichen. A model based approach to combine load and functional tests for service oriented architectures. In *VECoS*, pages 123–140, 2016.

[168] Afef Jmal Maâlej, Moez Krichen, and Mohamed Jmaiel. Conformance testing of ws-bpel compositions under various load conditions. In *2012 IEEE 36th annual computer software and applications conference*, pages 371–371. IEEE, 2012.

[169] Afef Jmal Maâlej, Moez Krichen, and Mohamed Jmaiel. Model-based conformance testing of ws-bpel compositions. In *2012 IEEE 36th annual computer software and applications conference workshops*, pages 452–457. IEEE, 2012.

[170] Afef Jmal Maâlej, Mariam Lahami, Moez Krichen, and Mohamed Jmaïel. Distributed and resource-aware load testing of ws-bpel compositions. In *ICEIS (2)*, pages 29–38, 2018.

[171] Saraswati Maddala and Sonali Patil. Agentless automation model for post exploitation penetration testing. In *Intelligent Computing, Information and Control Systems : ICICCS 2019*, pages 529–539. Springer, 2020.

[172] AV Shreyas Madhav, A Mohan, and Amit Kumar Tyagi. Improve : Intelligent machine learning based portable, reliable and optimal verification system for future vehicles. In *2023 International Conference on Computer Communication and Informatics (ICCCI)*, pages 1–6. IEEE, 2023.

[173] Shahid Mahmood, Alexy Fouillade, Hoang Nga Nguyen, and Siraj A Shaikh. A model-based security testing approach for automotive over-the-air updates. In *2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW)*, pages 6–13. IEEE, 2020.

[174] Shahid Mahmood, Hoang Nga Nguyen, and Siraj Ahmed Shaikh. Systematic threat assessment and security testing of automotive over-the-air (ota) updates. *Vehicular Communications*, 35 :100468, 2022.

[175] Jahanzaib Malik and Fabrizio Pastore. An empirical study of vulnerabilities in edge frameworks to support security testing improvement. *Empirical Software Engineering*, 28(4) :99, 2023.

[176] Raluca Marinescu. *Model-checking and model-based testing of automotive embedded systems : Starting from the system architecture*. PhD thesis, Mälardalen University, 2014.

[177] Arooj Masood, Demeke Shumeye Lakew, and Sungrae Cho. Security and privacy challenges in connected vehicular cloud computing. *IEEE Communications Surveys & Tutorials*, 22(4) :2725–2764, 2020.

[178] Francesc Mateo Tudela, Juan-Ramon Bermejo Higuera, Javier Bermejo Higuera, Juan-Antonio Sicilia Montalvo, and Michael I Argyros. On combining static, dynamic and interactive analysis security testing tools to improve owasp top ten security vulnerability detection in web applications. *Applied Sciences*, 10(24) :9119, 2020.

[179] Dean Richard McKinnel, Tooska Dargahi, Ali Dehghantanha, and Kim-Kwang Raymond Choo. A systematic literature review and meta-analysis on artificial intelligence in penetration testing and vulnerability assessment. *Computers & Electrical Engineering*, 75 :175–188, 2019.

[180] Na Meng, Stefan Nagy, Danfeng Yao, Wenjie Zhuang, and Gustavo Arango Argoty. Secure coding practices in java : Challenges and vulnerabilities. In *Proceedings of the 40th International Conference on Software Engineering*, pages 372–383, 2018.

[181] Saeed Mian Qaisar, Nehal Alyamani, Asad Waqar, and Moez Krichen. Machine learning with adaptive rate processing for power quality disturbances identification. *SN Computer Science*, 3 :1–6, 2022.

[182] Jan Midtgaard and Anders Møller. Quickchecking static analysis properties. *Software Testing, Verification and Reliability*, 27(6) :e1640, 2017.

[183] Alaeddine Mihoub, Moez Krichen, Mohannad Alswailim, Sami Mahfoudhi, and Riadh Bel Hadj Salah. Road scanner : A road state scanning approach based on machine learning techniques. *Applied Sciences*, 13(2) :683, 2023.

[184] Alaeddine Mihoub, Hosni Snoun, Moez Krichen, Riadh Bel Hadj Salah, and Montassar Kahia. Predicting covid-19 spread level using socio-economic indicators and machine learning techniques. In *2020 first international conference of smart systems and emerging technologies (SMARTTECH)*, pages 128–133. IEEE, 2020.

[185] Charlie Miller and Chris Valasek. A survey of remote automotive attack surfaces. *black hat USA*, 2014 :94, 2014.

[186] Negin Moghadasi, Amar Kulkarni, Dustin Crayton, Robert Grissom, James H Lambert, and Lu Feng. Formal methods in unmanned aerial vehicle swarm control for wildfire detection and monitoring. In *2023 IEEE International Systems Conference (SysCon)*, pages 1–8. IEEE, 2023.

[187] Sina Mohseni, Mandar Pitale, Vasu Singh, and Zhangyang Wang. Practical solutions for machine learning safety in autonomous vehicles. *arXiv preprint arXiv :1912.09630*, 2019.

[188] Abdul Moiz and Manar H Alalfi. A survey of security vulnerabilities in android automotive apps. In *Proceedings of the 3rd International Workshop on Enginee-*

*ring and Cybersecurity of Critical Systems*, pages 17–24, 2022.

[189] Muhammad Baqer Mollah, Jun Zhao, Dusit Niyato, Yong Liang Guan, Chau Yuen, Sumei Sun, Kwok-Yan Lam, and Leong Hai Koh. Blockchain for the internet of vehicles towards intelligent transportation systems : A survey. *IEEE Internet of Things Journal*, 8(6) :4157–4185, 2020.

[190] Lama J Moukahal, Mohammad Zulkernine, and Martin Soukup. Vulnerability-oriented fuzz testing for connected autonomous vehicle systems. *IEEE Transactions on Reliability*, 70(4) :1422–1437, 2021.

[191] Fabiola Moyón, Pamela Almeida, Daniel Riofrío, Daniel Mendez, and Marcos Kalinowski. Security compliance in agile software development : a systematic mapping study. In *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 413–420. IEEE, 2020.

[192] Philipp Mundhenk, Sebastian Steinhorst, Martin Lukasiewycz, Suhaib A Fahmy, and Samarjit Chakraborty. Security analysis of automotive architectures using probabilistic model checking. In *Proceedings of the 52nd Annual Design Automation Conference*, pages 1–6, 2015.

[193] Hussam Saeed Musa, Moez Krichen, Adem Alpaslan Altun, and Meryem Ammi. Survey on blockchain-based data storage security for android mobile applications. *Sensors*, 23(21) :8749, 2023.

[194] Tarak Nandy, Mohd Yamani Idna Bin Idris, Rafidah Md Noor, Laiha Mat Kiah, Lau Sian Lun, Nor Badrul Annuar Juma'at, Ismail Ahmedy, Norjihan Abdul Ghani, and Sananda Bhattacharyya. Review on security of internet of things authentication mechanism. *IEEE Access*, 7 :151054–151089, 2019.

[195] Nguyen Xuan Nhu, To Trong Nghia, Nguyen Huu Quyen, Van-Hau Pham, Phan The Duy, et al. Leveraging deep reinforcement learning for automating penetration testing in reconnaissance and exploitation phase. In *2022 RIVF International Conference on Computing and Communication Technologies (RIVF)*, pages 41–46. IEEE, 2022.

[196] Tugsmandakh Nyamdelger, Munkhdelgerekh Batzorig, Esam Ali Albhelil, Yeji Koh, and Kangbin Yim. Fuzz testing and safe framework development for vehicle security analysis. In *International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing*, pages 103–111. Springer, 2023.

[197] Bentley James Oakes, Mehrdad Moradi, Simon Van Mierlo, Hans Vangheluwe, and Joachim Denil. Machine learning-based fault injection for hazard analysis and risk assessment. In *International Conference on Computer Safety, Reliability, and Security*, pages 178–192. Springer, 2021.

[198] Dennis Kengo Oka. *Building secure cars : assuring the automotive software development lifecycle*. John Wiley & Sons, 2021.

[199] Md Mehedi Hassan Onik, KIM Chul-Soo, and YANG Jinhong. Personal data privacy challenges of the fourth industrial revolution. In *2019 21st International Conference on Advanced Communication Technology (ICACT)*, pages 635–638. IEEE, 2019.

[200] Seunghyun Park and Jin-Young Choi. Malware detection in self-driving vehicles using machine learning algorithms. *Journal of advanced transportation*, 2020 :1–9, 2020.

[201] Pranav Patki, Ajey Gotkhindikar, and Sunil Mane. Intelligent fuzz testing framework for finding hidden vulnerabilities in automotive environment. In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pages 1–4. IEEE, 2018.

[202] Rajshakhar Paul. Astor : An approach to identify security code reviews. In *Proceedings of the 37th IEEE/ACM International Conference on Automated Software Engineering*, pages 1–3, 2022.

[203] Irdin Pekaric, Clemens Sauerwein, and Michael Felderer. Applying security testing techniques to automotive engineering. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, pages 1–10, 2019.

[204] Irdin Pekaric, Clemens Sauerwein, Stefan Haselwanter, and Michael Felderer. A taxonomy of attack mechanisms in the automotive domain. *Computer Standards & Interfaces*, 78 :103539, 2021.

[205] Michele Peroli, Federico De Meo, Luca Viganò, and Davide Guardini. Mobster : A model-based security testing framework for web applications. *Software Testing, Verification and Reliability*, 28(8) :e1685, 2018.

[206] Steffen Pfrang, David Meier, and Valentin Kautz. Towards a modular security testing framework for industrial automation and control systems : Isutest. In *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–5. IEEE, 2017.

[207] Minh Pham and Kaiqi Xiong. A survey on security attacks and defense techniques for connected and autonomous vehicles. *Computers & Security*, 109 :102269, 2021.

[208] Sasanka Potluri and Christian Diedrich. High performance intrusion detection and prevention systems : a survey. In *ECCWS2016-Proceedings fo the 15th European Conference on Cyber Warfare and Security*, page 260. Academic Conferences and publishing limited, 2016.

[209] Adnan Qayyum, Muhammad Usama, Junaid Qadir, and Ala Al-Fuqaha. Securing connected & autonomous vehicles : Challenges posed by adversarial machine learning and the way forward. *IEEE Communications Surveys & Tutorials*, 22(2) :998–1026, 2020.

[210] Jochen Quante. Use cases of a generic model interpreter in an automotive software setting. In *2016 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 539–542. IEEE, 2016.

[211] Md Abdur Rahim, Md Arafatur Rahman, Md Mustafizur Rahman, A Taufiq Asyhari, Md Zakirul Alam Bhuiyan, and D Ramasamy. Evolution of iot-enabled connectivity and applications in automotive industry : A review. *Vehicular Communications*, 27 :100285, 2021.

[212] Nijat Rajabli, Francesco Flammini, Roberto Nardone, and Valeria Vittorini. Software verification and validation of safe autonomous cars : A systematic literature

review. *IEEE Access*, 9 :4797–4819, 2020.

[213] Shalli Rani, Ali Kashif Bashir, Moez Krichen, Abdulaziz Alshammari, et al. A low-rank learning based multi-label security solution for industry 5.0 consumers using machine learning classifiers. *IEEE Transactions on Consumer Electronics*, 2023.

[214] Iva Ranjan and Ram Bhushan Agnihotri. Ambiguity in cloud security with malware-injection attack. In *2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 306–310. IEEE, 2019.

[215] Adnan Rashid and Osman Hasan. Formal analysis of linear control systems using theorem proving. In *Formal Methods and Software Engineering : 19th International Conference on Formal Engineering Methods, ICFEM 2017, Xi'an, China, November 13-17, 2017, Proceedings*, pages 345–361. Springer, 2017.

[216] Adnan Rashid, Osman Hasan, and Sa'ed Abed. Using an interactive theorem prover for formally analyzing the dynamics of the unmanned aerial vehicles. In *Mobile Robot : Motion Control and Path Planning*, pages 253–282. Springer, 2023.

[217] Geetanjali Rathee, Ashutosh Sharma, Razi Iqbal, Moayad Aloqaily, Naveen Jaglan, and Rajiv Kumar. A blockchain framework for securing connected and autonomous vehicles. *Sensors*, 19(14) :3165, 2019.

[218] Kotha Raj Kumar Reddy, Angappa Gunasekaran, P Kalpana, V Raja Sreedharan, and S Arvind Kumar. Developing a blockchain framework for the automotive supply chain : A systematic review. *Computers & Industrial Engineering*, 157 :107334, 2021.

[219] Thomas Reps and Aditya Thakur. Automating abstract interpretation. In *Verification, Model Checking, and Abstract Interpretation : 17th International Conference, VMCAI 2016, St. Petersburg, FL, USA, January 17-19, 2016. Proceedings 17*, pages 3–40. Springer, 2016.

[220] Indranil Roy, Shekhar Sonthalia, Trideep Mandal, Animesh Kairi, and Mohuya Chakraborty. Study on network scanning using machine learning-based methods. In *Proceedings of International Ethical Hacking Conference 2019 : eHaCON 2019, Kolkata, India*, pages 77–85. Springer, 2020.

[221] Marcel Rumez, Daniel Grimm, Reiner Kriesten, and Eric Sax. An overview of automotive service-oriented architectures and implications for security countermeasures. *IEEE access*, 8 :221852–221870, 2020.

[222] Michael L Rustad and Thomas H Koenig. Towards a global data privacy standard. *Fla. L. Rev.*, 71 :365, 2019.

[223] Memoona Sadaf, Zafar Iqbal, Abdul Rehman Javed, Irum Saba, Moez Krichen, Sajid Majeed, and Arooj Raza. Connected and automated vehicles : Infrastructure, applications, security, critical challenges, and future aspects. *Technologies*, 11(5) :117, 2023.

[224] Dhaou Said, Mayssa Elloumi, and Lyes Khoukhi. Cyber-attack on p2p energy transaction between connected electric vehicles : A false data injection detection based machine learning model. *IEEE Access*, 10 :63640–63647, 2022.

[225] Siwar Ben Hadj Said, Bernard Cousin, and Samer Lahoud. Software defined networking (sdn) for reliable user connectivity in 5g networks. In *2017 IEEE Conference on Network Softwarization (NetSoft)*, pages 1–5. IEEE, 2017.

[226] Nadir K Salih, D Satyanarayana, Abdullah Said Alkalbani, and R Gopal. A survey on software/hardware fault injection tools and techniques. In *2022 IEEE Symposium on Industrial Electronics & Applications (ISIEA)*, pages 1–7. IEEE, 2022.

[227] Muhammad Salman Sarfraz, Hyunsoo Hong, and Seong Su Kim. Recent developments in the manufacturing technologies of composite components and their cost-effectiveness in the automotive industry : A review study. *Composite Structures*, 266 :113864, 2021.

[228] Dalila Say, Salah Zidi, Saeed Mian Qaisar, and Moez Krichen. Automated categorization of multiclass welding defects using the x-ray image augmentation and convolutional neural network. *Sensors*, 23(14) :6422, 2023.

[229] Mohd Abuzar Sayeed, Mohd Asim Sayeed, and Sharad Saxena. Intrusion detection system based on software defined network firewall. In *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, pages 379–382. IEEE, 2015.

[230] Christoph Schmittner and Georg Macher. Automotive cybersecurity standards-relation and overview. In *Computer Safety, Reliability, and Security : SAFECOMP 2019 Workshops, ASSURE, DECSoS, SASSUR, STRIVE, and WAISE, Turku, Finland, September 10, 2019, Proceedings 38*, pages 153–165. Springer, 2019.

[231] Stefan Schönhärl, Philipp Fuxen, Julian Graf, Jonas Schmidt, Rudolf Hackenberg, and Jürgen Mottok. An automotive penetration testing framework for it-security education. *CLOUD COMPUTING 2022*, page 10, 2022.

[232] Amar Seeam, Ochanya S Ogbeh, Shivanand Guness, and Xavier Bellekens. Threat modeling and security issues for the internet of things. In *2019 conference on next generation computing applications (NextComp)*, pages 1–8. IEEE, 2019.

[233] Collins Sey, Hang Lei, Weizhong Qian, Xiaoyu Li, Linda Delali Fiasam, Seth Larweh Kodjiku, Isaac Adjei-Mensah, and Isaac Osei Agyemang. Vblock : A blockchain-based tamper-proofing data protection model for internet of vehicle networks. *Sensors*, 22(20) :8083, 2022.

[234] Pradip Kumar Sharma, Neeraj Kumar, and Jong Hyuk Park. Blockchain-based distributed framework for automotive industry in a smart city. *IEEE Transactions on Industrial Informatics*, 15(7) :4197–4205, 2018.

[235] Shaila Sharmin and Hafizah Mansor. Intrusion detection on the in-vehicle network using machine learning. In *2021 3rd International Cyber Resilience Conference (CRC)*, pages 1–6. IEEE, 2021.

[236] Rajesh Shrivastava, Simar Preet Singh, and Mohammad Kamrul Hasan. Code tamper-proofing using return oriented programming in iot devices. In *Rising Threats in Expert Applications and Solutions : Proceedings of FICR-TEAS 2022*, pages 167–174. Springer, 2022.

[237] Rajesh Kumar Shrivastava, Simar Preet Singh, Mohammad Kamrul Hasan, Shayla Islam, Salwani Abdullah, Azana Hafizah Mohd Aman, et al. Securing internet of things devices against code tampering attacks using return oriented programming. *Computer Communications*, 193 :38–46, 2022.

[238] Amit Mazumder Shuvo, Nitin Pundir, Jungmin Park, Farimah Farahmandi, and Mark Tehranipoor. Ldtfi : Layout-aware timing fault-injection attack assessment against differential fault analysis. In *2022 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 134–139. IEEE, 2022.

[239] Larry Singleton, Rui Zhao, Myoungkyu Song, and Harvey Siy. Cryptotutor : Teaching secure coding practices through misuse pattern detection. In *Proceedings of the 21st Annual Conference on Information Technology Education*, pages 403–408, 2020.

[240] Sepha Siswantyo. Security analysis and improvement of lighweight vanet authentication protocol (case study : Zhao et al. lvap). *Journal of Computer Networks, Architecture and High Performance Computing*, 3(2) :135–143, 2021.

[241] Steven So, Prinkle Sharma, and Jonathan Petit. Integrating plausibility checks and machine learning for misbehavior detection in vanet. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 564–571. IEEE, 2018.

[242] Florian Sommer, Jürgen Dürrwang, and Reiner Kriesten. Survey and classification of automotive security attacks. *Information*, 10(4) :148, 2019.

[243] Florian Sommer, Reiner Kriesten, and Frank Kargl. Survey of model-based security testing approaches in the automotive domain. *IEEE Access*, 2023.

[244] Marcelo Sousa, César Rodríguez, Vijay D'Silva, and Daniel Kroening. Abstract interpretation with unfoldings. In *Computer Aided Verification : 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part II 30*, pages 197–216. Springer, 2017.

[245] Ivan Studnia, Vincent Nicomette, Eric Alata, Yves Deswarte, Mohamed Kaâniche, and Youssef Laarouchi. Survey on security threats and protection mechanisms in embedded automotive networks. In *2013 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W)*, pages 1–12. IEEE, 2013.

[246] Peng Su and DeJiu Chen. Using fault injection for the training of functions to detect soft errors of dnns in automotive vehicles. In *International Conference on Dependability and Complex Systems*, pages 308–318. Springer, 2022.

[247] Zhou Su, Yuntao Wang, Qichao Xu, Minrui Fei, Yu-Chu Tian, and Ning Zhang. A secure charging scheme for electric vehicles with smart communities in energy blockchain. *IEEE Internet of Things Journal*, 6(3) :4601–4613, 2018.

[248] Peter Subke, Muzafar Moshref, Andreas Vach, and Markus Steffelbauer. Measures to prevent unauthorized access to the in-vehicle e/e system, due to the security vulnerability of a remote diagnostic tester. *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, 10(2017-01-1689) :422–429, 2017.

[249] Xiaoqiang Sun, F Richard Yu, and Peng Zhang. A survey on cyber-security of connected and autonomous vehicles (cavs). *IEEE Transactions on Intelligent Transportation Systems*, 23(7) :6240–6259, 2021.

[250] Yixin Sun, Kangkook Jee, Suphannee Sivakorn, Zhichun Li, Cristian Lumezanu, Lauri Korts-Parn, Zhenyu Wu, Junghwan Rhee, Chung Hwan Kim, Mung Chiang, et al. Detecting malware injection with program-dns behavior. In *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 552–568. IEEE, 2020.

[251] Devin Sweigert, Md Minhaz Chowdhury, and Nafiz Rifat. Exploit security vulnerabilities by penetration testing. In *2022 IEEE International Conference on Electro Information Technology (eIT)*, pages 527–532. IEEE, 2022.

[252] Mohammad Tahaei and Kami Vaniea. A survey on developer-centred security. In *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 129–138. IEEE, 2019.

[253] Fengxiao Tang, Yuichi Kawamoto, Nei Kato, and Jiajia Liu. Future intelligent and secure vehicular network toward 6g : Machine-learning approaches. *Proceedings of the IEEE*, 108(2) :292–307, 2019.

[254] Christopher Thompson and David Wagner. A large-scale study of modern code review and security in open source projects. In *Proceedings of the 13th International Conference on Predictive Models and Data Analytics in Software Engineering*, pages 83–92, 2017.

[255] Vassil Todorov, Frédéric Boulanger, and Safouan Taha. Formal verification of automotive embedded software. In *Proceedings of the 6th Conference on Formal Methods in Software Engineering*, pages 84–87, 2018.

[256] Michal Trnka, Tomas Cerny, and Nathaniel Stickney. Survey of authentication and authorization for the internet of things. *Security and Communication Networks*, 2018, 2018.

[257] Aashma Uprety, Danda B Rawat, and Jiang Li. Privacy preserving misbehavior detection in iov using federated machine learning. In *2021 IEEE 18th annual consumer communications & networking conference (CCNC)*, pages 1–6. IEEE, 2021.

[258] Nazar Waheed, Xiangjian He, Muhammad Ikram, Muhammad Usman, Saad Sajid Hashmi, and Muhammad Usman. Security and privacy in iot using machine learning and blockchain : Threats and countermeasures. *ACM Computing Surveys (CSUR)*, 53(6) :1–37, 2020.

[259] Timothy Werquin, Mathijs Hubrechtsen, Ashok Thangarajan, Frank Piessens, and Jan Tobias Mühlberg. Automated fuzzing of automotive control units. In *2019 International Workshop on Secure Internet of Things (SIOT)*, pages 1–8. IEEE, 2019.

[260] Wenjun Xiong, Fredrik Krantz, and Robert Lagerström. Threat modeling and attack simulations of connected vehicles : Proof of concept. In *Information Systems Security and Privacy : 5th International Conference, ICISSP 2019, Prague, Czech Republic, February 23-25, 2019, Revised Selected Papers 5*, pages 272–287.

Springer, 2020.

[261] Wenjun Xiong and Robert Lagerström. Threat modeling–a systematic literature review. *Computers & security*, 84 :53–69, 2019.

[262] Wenjun Xiong, Emeline Legrand, Oscar Åberg, and Robert Lagerström. Cyber security threat modeling based on the mitre enterprise att&ck matrix. *Software and Systems Modeling*, 21(1) :157–177, 2022.

[263] Yuan Xu, Xujie Li, Mengran Jin, and Yong Lu. A trusted distribution mechanism of tasks for the internet of vehicles based on blockchain. In *2021 13th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–5. IEEE, 2021.

[264] Yijie Xun, Jiajia Liu, Nei Kato, Yongqiang Fang, and Yanning Zhang. Automobile driver fingerprinting : A new machine learning based authentication scheme. *IEEE Transactions on Industrial Informatics*, 16(2) :1417–1426, 2019.

[265] Tomoya Yamaguchi, Martin Brain, Chirs Ryder, Yosikazu Imai, and Yoshiumi Kawamura. Application of abstract interpretation to the automotive electronic control system. In *Verification, Model Checking, and Abstract Interpretation : 20th International Conference, VMCAI 2019, Cascais, Portugal, January 13–15, 2019, Proceedings 20*, pages 425–445. Springer, 2019.

[266] Tomoya Yamaguchi, Tomoyuki Kaga, Alexandre Donzé, and Sanjit A Seshia. Combining requirement mining, software model checking and simulation-based verification for industrial automotive systems. In *2016 Formal Methods in Computer-Aided Design (FMCAD)*, pages 201–204. IEEE, 2016.

[267] Abel Yeboah-Ofori and Shareeful Islam. Cyber security threat modeling for supply chain organizational environments. *Future internet*, 11(3) :63, 2019.

[268] Junkai Yi and Xiaoyan Liu. Deep reinforcement learning for intelligent penetration testing path design. *Applied Sciences*, 13(16) :9467, 2023.

[269] Clinton Young, Joseph Zambreno, Habeeb Olufowobi, and Gedare Bloom. Survey of automotive controller area network intrusion detection systems. *IEEE Design & Test*, 36(6) :48–55, 2019.

[270] Zhang Yu, Syed Abdul Rehman Khan, and Muhammad Umar. Circular economy practices and industry 4.0 technologies : A strategic move of automobile industry. *Business Strategy and the Environment*, 31(3) :796–809, 2022.

[271] Ilke Yurtseven and Selami Bagriyanik. A review of penetration testing and vulnerability assessment in cloud environment. In *2020 Turkish National Software Engineering Symposium (UYMS)*, pages 1–6. IEEE, 2020.

[272] Daniel Zelle, Timm Lauser, Dustin Kern, and Christoph Krauß. Analyzing and securing some/ip automotive services with formal and practical methods. In *Proceedings of the 16th International Conference on Availability, Reliability and Security*, pages 1–20, 2021.

[273] Yong Zeng, Rui Zhang, and Teng Joon Lim. Wireless communications with unmanned aerial vehicles : Opportunities and challenges. *IEEE Communications magazine*, 54(5) :36–42, 2016.

[274] Haichun Zhang, Kelin Huang, Jie Wang, and Zhenglin Liu. Can-ft : A fuzz testing method for automotive controller area network bus. In *2021 International Conference on Computer Information Science and Artificial Intelligence (CISAI)*, pages 225–231. IEEE, 2021.

[275] Maoshen Zhang, He Li, Peijing Wang, and Qiang Liu. Parity check based fault detection against timing fault injection attacks. *Electronics*, 11(24) :4082, 2022.

[276] Zhirui Zhang, Dave Towey, Zhihao Ying, Yifan Zhang, and Zhi Quan Zhou. Mt4ns : Metamorphic testing for network scanning. In *2021 IEEE/ACM 6th International Workshop on Metamorphic Testing (MET)*, pages 17–23. IEEE, 2021.

[277] Cheng Zhou, Hanbin Luo, Weili Fang, Ran Wei, and Lieyun Ding. Cyber-physical-system-based safety monitoring for blind hoisting with the internet of things : A case study. *Automation in construction*, 97 :138–150, 2019.

[278] Salah Zidi, Alaeddine Mihoub, Saeed Mian Qaisar, Moez Krichen, and Qasem Abu Al-Haija. Theft detection dataset for benchmarking and machine learning based classification in a smart grid environment. *Journal of King Saud University-Computer and Information Sciences*, 35(1) :13–25, 2023.

[279] Anton Zita, Sahar Mohajerani, and Martin Fabian. Application of formal verification to the lane change module of an autonomous vehicle. In *2017 13th IEEE Conference on Automation Science and Engineering (CASE)*, pages 932–937. IEEE, 2017.

[280] Ioannis Zografopoulos, Juan Ospina, Xiaorui Liu, and Charalambos Konstantinou. Cyber-physical energy systems security : Threat modeling, risk assessment, resources, metrics, and case studies. *IEEE Access*, 9 :29775–29818, 2021.